

2015

An information theoretic approach for generating an aircraft avoidance Markov decision process

<https://hdl.handle.net/2144/15208>

Boston University

BOSTON UNIVERSITY
COLLEGE OF ENGINEERING

Thesis

**AN INFORMATION THEORETIC APPROACH FOR
GENERATING AN AIRCRAFT AVOIDANCE MARKOV
DECISION PROCESS**

by

ANDREW J. WEINERT

B.S., Pennsylvania State University, 2009

Submitted in partial fulfillment of the
requirements for the degree of
Master of Science

2015

© 2015 by
ANDREW J. WEINERT
All rights reserved

Approved by

First Reader



David A. Castañón, PhD
Professor of Electrical and Computer Engineering
Professor of Systems Engineering

Second Reader



Michael P. Owen, PhD
Technical Staff, Surveillance Systems Group
Lincoln Laboratory
Massachusetts Institute of Technology

Third Reader



Ioannis Ch. Paschalidis
Professor of Electrical and Computer Engineering
Professor of Systems Engineering

Aut viam inveniam aut faciam;

Hannibal

Acknowledgments

This work is sponsored under Air Force Contract #FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

Thank you Dr. Castañón for giving me the flexibility to explore, the composure of bring me down to earth, and insight into conceptualizing very complex problems.

Thank you Dr. Michael Owen for pushing me to stay on pace, giving me room to fail, and righting my course when I've lost some common sense.

Thank you Robert Klaus for letting me think out loud my problems to you and providing valuable insight and understanding into the inner workings of ACAS X.

Thank you Dr. Mykel Kochenderfer for accepting as an intern many moons ago, which has led me to this thesis and for making sure I never settle on anything.

Thank you Stephanie Cipolla for your support, patience, and encouragement, along with all the quality time we spent together in the library.

**AN INFORMATION THEORETIC APPROACH FOR
GENERATING AN AIRCRAFT AVOIDANCE MARKOV
DECISION PROCESS
ANDREW J. WEINERT**

ABSTRACT

Developing a collision avoidance system that can meet safety standards required of commercial aviation is challenging. A dynamic programming approach to collision avoidance has been developed to optimize and generate logics that are robust to the complex dynamics of the national airspace. The current approach represents the aircraft avoidance problem as Markov Decision Processes and independently optimizes a horizontal and vertical maneuver avoidance logics. This is a result of the current memory requirements for each logic, simply combining the logics will result in a significantly larger representation. The “curse of dimensionality” makes it computationally inefficient and infeasible to optimize this larger representation. However, existing and future collision avoidance systems have mostly defined the decision process by hand.

In response, a simulation-based framework was built to better understand how each potential state quantifies the aircraft avoidance problem with regards to safety and operational components. The framework leverages recent advances in signals processing and database, while enabling the highest fidelity analysis of Monte Carlo aircraft encounter simulations to date. This framework enabled the calculation of how well each state of the decision process quantifies the collision risk and the associated memory requirements. Using this analysis, a collision avoidance logic that leverages both horizontal and vertical actions was built and optimized using this simulation-based approach.

Contents

1	Introduction	1
1.1	Motivation	3
1.2	Objectives	5
1.3	Contributions	6
1.4	Outline	8
2	Background	9
2.1	Markov Decision Process Optimization	9
2.1.1	Solving MDPs	11
2.1.2	Shape Optimization	12
2.2	Computational Considerations	14
2.2.1	Storage and Access	14
2.2.2	Matrix Representation	16
2.2.3	Entropy	18
2.3	Aircraft Avoidance Algorithms	19
2.3.1	TCAS	22
2.3.2	Horizontal TCAS	25
2.3.3	ACAS X	26
2.3.4	Non-Dynamic Programming Based	31
2.4	Encounter Models	33
2.4.1	Types and Use	34
2.4.2	Development and Structure	36

3	Theory and Approach	39
3.1	Overview	39
3.2	Conceptualizing the Aircraft Avoidance Problem	41
3.2.1	Safety	42
3.2.2	Operational	44
3.3	NMAC Entropy	45
3.4	Action Space	46
3.5	State Space	50
3.5.1	System Perspective	53
3.5.2	Vertical Axis	54
3.5.3	Horizontal Plane	58
3.6	MDP Cost	64
3.6.1	NMAC Horizon	64
3.6.2	Cost Functions	66
3.7	Policy Evaluation	68
4	Implementation	71
4.1	Monte Carlo Simulations	72
4.2	Simulation Processing	74
4.3	Calculate state-transition matrix	74
4.4	Generate cost matrix	76
4.5	Optimize policy	77
4.6	Evaluate policy	78
5	Results and Evaluation	79
5.1	State-transition generation	79
5.2	State Entropy	82
5.2.1	Separation	82

5.2.2	Range Rates	85
5.2.3	Simple Tau	86
5.2.4	Angles	90
5.2.5	Airspeed	91
5.2.6	Vertical Rates	92
5.3	Identifying NMAC Risk	95
5.3.1	Baseline $\{r_h, \Delta h\}$ State Space	97
5.3.2	Adding States	102
5.4	Policy Generation	112
5.4.1	Baseline	112
5.4.2	Adding States using Information Theory	114
5.4.3	Higher order states spaces	117
5.4.4	Other Considerations	120
5.5	Evaluation Results	123
5.5.1	Baseline $\{r_h, \Delta h\}$	124
5.5.2	Effect of adding range rate	127
5.5.3	Full state space	129
6	Conclusion	132
6.1	Contributions of Simulation-Framework	133
6.1.1	Memory management	133
6.1.2	State and action selection	134
6.1.3	Delay and RA state	135
6.2	Contributions of Information Theoretics	135
6.2.1	Quantifying risk of NMAC	136
6.2.2	Surrogate for MDP state's utility	136
6.3	Future Work	137

A	Simulation States	139
B	Encounter Geometry Types	141
B.1	Vertical	141
B.2	Horizontal	142
	References	144
	Curriculum Vitae	153

List of Tables

2.1	TCAS correction resolution advisory (RA) action set.	24
2.2	Encounter model categories.	36
3.1	Action set of representative manned aircraft.	50
3.2	Action set of representative unmanned aircraft system (UAS).	50
5.1	Memory requirements for random state-transition matrices.	82
5.2	Baseline $\{r_h, \Delta h\}$ results with manned action space	124
5.3	$\{r_h, \Delta h, \dot{r}_h \mid C_{\rho_h} = C_{\rho_v}\}$ results with manned action space	127
5.4	$\left\{r_h, \Delta h, \dot{r}_h, \alpha, \left(\dot{h}_o + \dot{h}_i\right), s_{RA} \mid C_{\rho_h} = C_{\rho_v}\right\}$ results	130

List of Figures

1·1	ACAS X trajectory propagation methods.	2
2·1	Memory required for MATLAB $10,000 \times 10,000$ matrix.	17
2·2	Development and evolution of airborne collision avoidance	21
2·3	Traditional collision avoidance and self separation layered architecture	22
2·4	TCAS states	23
2·5	TCAS altitude criterion.	24
2·6	Enhanced TCAS II horizontal algorithm	26
2·7	ACAS X logic development approach.	29
2·8	JOCA MPC maneuver selection hierarchy.	32
2·9	RADES sensor coverage map	37
3·1	Example aircraft vertical action variety	47
3·2	Required airspeed and bank angle for various turn rates.	49
3·3	Real number coordinate systems	51
3·4	Feature-based space aggregation.	54
3·5	First-order vertical states.	55
3·6	Simple visualization of vertical rate states	55
3·7	$(\dot{h}_o + \dot{h}_i \mid \dot{h}_o, \dot{h}_i)$ distribution and CDF	56
3·8	$\Delta h(r_s, \theta_s)$ distribution and CDF	58
3·9	Range rate of head-on encounter given airspeed	59
3·10	$\tau_h(v_o, v_i \mid r_h = 30381)$ distribution and CDF	60
3·11	Simple visualization of angular states	62

3·12	Example encounter	62
3·13	Various states for example encounter in Figure 3·12	63
4·1	Architecture of an aircraft in CASSATT	72
4·2	$\Delta h(s, s')$ state transitions with naive cost matrix	76
4·3	CASSATT evaluation framework	78
5·1	Convex hull of $\Delta h(s, s')$ using manned action set	80
5·2	PDF of $\Delta h(s)$ using manned action set	81
5·3	Separation states: $\Delta h, r_h, r_s$	83
5·4	Entropy and memory of individual COC range states	83
5·5	Entropy (bits) of COC $\{r_h, \Delta h\}$	85
5·6	Memory (MB) of COC $\{r_h, \Delta h\}$	85
5·7	Entropy and memory of range rate states	86
5·8	Entropy and memory of COC simple τ states	87
5·9	NMAC entropy and probability distribution for $\tau_{h,100}(s)$	88
5·10	Entropy and probability distribution for COC $\tau_{h,100}(s)$	89
5·11	Entropy and memory of COC angle states	91
5·12	Spherical perspective for a constant Δh	92
5·13	Entropy and memory of COC airspeed states	93
5·14	Entropy and memory of COC vertical rate states	94
5·15	Entropy and probability distribution for COC $(\dot{h}_o + \dot{h}_i)$	95
5·16	Entropy (bits) of COC $\{\Delta h, \dot{h}_o\}$ and $\{\Delta h, \dot{h}_o + \dot{h}_i\}$	96
5·17	Memory (MB) of COC $\{\Delta h, \dot{h}_o\}$ and $\{\Delta h, \dot{h}_o + \dot{h}_i\}$	96
5·18	NMAC horizons using manned action set for $\{r_h, \Delta h\}$	98
5·19	NMAC horizons using UAS action set for $\{r_h, \Delta h\}$	99
5·20	Vertical perspective within regards to action	100
5·21	Comparison of $\nu_{30} \{r_h, \Delta h\}$ convex hulls	101

5.22	Comparison of specific action ν_{30} $\{r_h, \Delta h\}$ convex hulls	101
5.23	Comparison of ν_{30} $\{r_h, \Delta h, \dot{\Delta h} \mid \text{CL1500/750}\}$	103
5.24	ν_{45} of manned action set for $\{r_h, \Delta h, \dot{\Delta h}\}$	104
5.25	ν_{45} of UAS action set for $\{r_h, \Delta h, \dot{\Delta h}\}$	105
5.26	ν_{45} of manned action set for $\{r_h, \Delta h, \dot{r}_h\}$	107
5.27	ν_{45} of UAS action set for $\{r_h, \Delta h, \dot{r}_h\}$	108
5.28	ν_{45} of manned action set for $\{r_h, \Delta h, \alpha\}$	110
5.29	ν_{45} of UAS action set for $\{r_h, \Delta h, \alpha\}$	111
5.30	$\{r_h, \Delta h\}$ policy using a ν_{30} -based alerting	112
5.31	$\{r_h, \Delta h\}$ policies of varying C_{ρ_h} using a ν_{30} -based alerting	113
5.32	Gain and memory increase of adding a state to $\{r_h, \Delta h\}$	115
5.33	$\{r_h, \Delta h, \dot{r}_h \mid C_{\rho_h} = C_{\rho_v}\}$ policies using ν_{30} alerting	117
5.34	Angular discretization vs. unique states	119
5.35	Required angular positions that the MDP must represent	120
5.36	Results using $\{r_h, \Delta h\}$	125
B.1	Vertical encounters	141
B.2	Head-on and overtaking encounters	142
B.3	Crossing and parallel encounters	142
B.4	Slow-closure encounter	143

List of Abbreviations

ACAS X Airborne Collision Avoidance System

ACID Atomicity, Consistency, Isolation, Durability

ADS-B automatic dependent surveillance-broadcast

ATC air traffic control

ATM air traffic management

BASE Basic Availability, Soft-state, Eventual consistency

BCAS Beacon Collision Avoidance System

CA collision avoidance

CASSATT Collision Avoidance System Safety Assessment Tool

COC clear of conflict

CONUS Contiguous United States

CPA closes point of approach

D4M Dynamic Distributed Dimensional Data Model

DP dynamic programming

DOD Department of Defense

FAA Federal Aviation Administration

ICAO International Civil Aviation Organization

IFR instrument flight rules

JOCA Jointly Optimal Collision Avoidance

LLGrid Lincoln Laboratory Grid

MDP Markov decision process

MPC model predictive control

MIAA Multiple Intruder Autonomous Avoidance

MuSICA Multi-sensor Integrated Conflict Avoidance

MIT LL MIT Lincoln Laboratory

NAS U.S National Airspace System

NMAC near mid-air collision

PDF probability density function

POMDP partially observed Markov decision process

RA resolution advisory

RADES Radar Evaluation Squadron

ROT rate one turn

RSM response surface methodology

SAA sense and avoid

SS self separation

TA traffic alert

TAS true airspeed

TCA time of closet approach

TCAS Traffic Alert and Collision Avoidance System

UAS unmanned aircraft system

VFR visual flight rules

VTOL vertical take-off and landing

Chapter 1

Introduction

The U.S National Airspace System (NAS) is a dynamic and complex system that supports conveniences of everyday life, the global economy, and the freedom to travel. Both its capacity and ability to maintain safety is stressed by numerous air-carriers, cargo airlines, business jets, general aviation, and, most recently, UAS. Regulatory organizations such as the Federal Aviation Administration (FAA) and the International Civil Aviation Organization (ICAO) mandate the use of manned collision avoidance systems to maintain an acceptable level of safety. It consists of sensor(s) to identify other intruder aircraft and an aircraft avoidance algorithm to recommend actions to avoid the intruders and prevent an imminent mid-air collision.

The first and to-date only globally mandated aircraft avoidance system is the Traffic Alert and Collision Avoidance System (TCAS) (William, 1989; Kuchar and Drumm, 2007). It warns pilots when they are in danger of a mid-air collision and has been mandated by the FAA on aircraft with at least ten seats. TCAS has been very successful in preventing mid-air collisions over the years through the use of vertical collision avoidance maneuvers. Yet, the way in which the logic was designed limits its robustness. Specifically, the tight integration between the system components and its sole functional design for manned collision avoidance. With the current evolution of the NAS, TCAS can no longer completely support the safety and operational requirements of the airspace. To meet these requirements, a major overhaul of TCAS and collision avoidance philosophy is required.

In response, the Airborne Collision Avoidance System (ACAS X) program is developing an aircraft avoidance logic represented as a numeric table that has been optimized with respect to models of the airspace (Holland et al., 2013). Using similar operational inputs and outputs as TCAS, ACAS X generates an avoidance algorithm via a logic optimization process using a probabilistic dynamic and multi-objective utility models. The probabilistic dynamical model is a statistical representation of where the aircraft will be in the future, and the multi-objective utility model represents the safety and operational objectives of the system. A Markov decision process (MDP) represents the dynamic model combined with the utility model. A numeric lookup table is generated via dynamic programming (DP) optimization. This contrasts the ad hoc rule-based pseudo code approach of TCAS. The ACAS X system reduces the number of alerts and increases safety when compared to the legacy TCAS system.

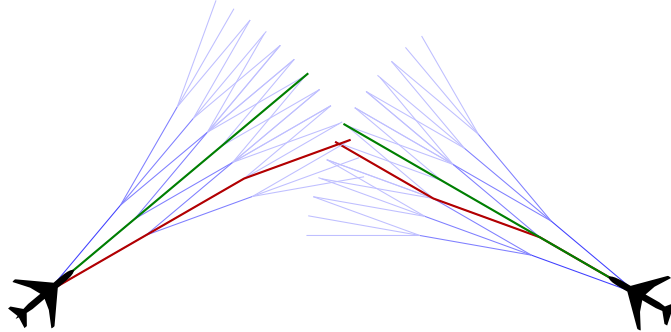


Figure 1-1: ACAS X trajectory propagation methods. Nominal trajectory propagation used by TCAS (green) only examines the most likely trajectory. Worst-case trajectory propagation (red) searches the space of possible trajectories that result in NMAC. Probabilistic trajectory propagation (blue) accounts for the relative likelihood of all trajectories.

The MDP DP approach of ACAS X has been endorsed by the FAA, which has designated it as TCAS's successor. The program has been expanded to develop new algorithms for UAS and general aviation, which are outside the historical user class of manned, commercial aircraft. These new algorithms may leverage horizontal maneuvers instead of the traditional vertical maneuvers. However an ACAS X algorithm

that considers both horizontal and vertical maneuvers separately has not developed. Horizontal and vertical maneuvers are currently considered independently. This is due to the consequences of the “curse of dimensionality (Powell, 2007),” a phenomena of high-dimensional problems. The direct consequence of adding dimensions is greater requirements computational power and memory for optimization. Attempts at developing a joint MDP through a union of the horizontal and vertical maneuver MDPs has proven to be computationally infeasible for DP optimization.

1.1 Motivation

The development of a joint horizontal and vertical maneuver MDP is important because many of the assumption about airborne collision avoidance are becoming irrelevant, outdated, or simply wrong in the new airspace. Supporting new user classes as UASs is a key operational requirement and is challenging because their performance characteristics can widely vary. Rotorcraft with their unique ability to hover currently lack a mandate or operational collision avoidance system. Due to the range of performances, it maybe faster to increase separation via a horizontal action. This is especially important for UAS where certain vertical rates can potentially interfere with communications. Examining the trade offs and relationships between different maneuvers will lead to the development of safer collision avoidance systems.

This challenge is compounded due the historical use of vertical only maneuvers in TCAS, there simply has been less research and development an evaluation of horizontal collision avoidance maneuvers. Due to growing need for new collision avoidance systems, insights and analysis of horizontal or joint maneuvers are becoming increasingly important. While the vertical maneuver variant of ACAS X can leverage most of the previous TCAS research and operational knowledge, there are less historical resources to assist in formulating and optimizing a horizontal or joint maneuver MDP.

For years, ACAS X has had a multi-organization team working on improving the optimization parameters and overall process. Many of these parameters are tailored for vertical maneuvers; they may not be suitable when considering both horizontal and vertical actions, limiting their utility of existing optimization configurations. This is attributed to a lack of a qualitative understanding on the relationship between horizontal and vertical maneuvers, better defining this will result in improvements in the optimization.

When developing these MDPs, there exists trade-offs between the robustness and completeness; computational requirements for optimization; and resulting operational safety of implementing the optimal MDP solution. Computation time is a finite resource and is an important factor in determining the feasibility of a system. Greater computation time results in fewer design iterations and longer development time. Additionally, if the optimal control solution requires significantly large memory requirements, it may not be operationally suitable. Aircraft avionic systems have finite memory resources and an aircraft avoidance algorithm must fit within memory. Deploying new avionic systems to enhance memory could result in costly certification and implementation problems (Kinnan, 2009). Furthermore the larger the memory requirements, the longer the potential read-time of the policy at runtime, which can subsequently introduce additional delay prior to maneuvers.

Understanding how aircraft behave during close encounters is critical when formulating the MDP. While probabilistic models that describe aircraft behavior exist, they are constrained to a selection of states and focus on manned aircraft. This limits their use when exploring different and new MDP formulations. A better understanding of the aircraft encounter space would lead to more efficient MDP formulations and allow the generation of a joint horizontal and vertical maneuver MDP for any aircraft user class. Additionally, a better understanding of aircraft encounter dynamics outside of

manned aircraft would provide critical insights to improve the safety of the NAS.

1.2 Objectives

Collision avoidance systems, like TCAS and ACAS X and assessments currently focus on a small set of historically validated states. These states have been selected by hand and directly correspond to physical dynamics than are easily observed by humans. Hence, one primary research objective is to develop and implement a quantitative analytic approach to identifying aircraft encounter states that are memory efficient and quantify the collision risk. This supports the other primary objective of formulating and optimizing an MDP that leverages both horizontal and vertical maneuvers; the analytic approach will guide the MDP state selection process.

To support these objectives, a new simulation-based framework was developed. Modifying the ACAS X framework and hard coding dynamics for each new potential state would be time consuming and inefficient. Instead a new simulation-based framework is implemented that leverages recent advances in signals processing and databases.

The analytic approach will apply information theoretic concepts to the aircraft avoidance problem. New metrics that quantify the information each state provides about a potential collision and that define the inherent collision risk for each state are required. Formulating new MDPs will require an insight into the trade offs and relationships between different maneuvers. Since ACAS X implements a traditional DP optimization, a new alternative optimization architecture that leverages simulation-based dynamics is required.

These research objectives focusing on demonstrating feasibility of the formulating MDP and the utility of applying information theoretics to collision avoidance. It is out of scope to include all ten TCAS vertical commands in addition to horizontal

commands, rather the emphasis is the development of MDP and the composition of the information vector, that describes the “state of the world.” Additionally, the emphasis is on the existence of an suitable MDP formulation and not the configuration of the DP optimization.

These objectives will be met if simulated large-scale aircraft data can be processed and used to formulate a MDP. The resulting collision avoidance algorithm will be evaluated using a historically-validated safety and operational metrics. With the emphasis on the evaluating of states and subsequent MDP formulation feasibility, the research objectives in order of importance are:

1. Develop a quantitative approach for selecting aircraft avoidance MDP states
2. Generate an aircraft collision avoidance algorithm that considers both horizontal and vertical actions
3. Implement a new simulation-based aircraft encounter analysis framework
4. Build an acuity on the relationship between horizontal and vertical maneuvers
5. Implement an alternative to ACAS X MDP optimization framework that allows greater flexibility in algorithm design

1.3 Contributions

The development of the simulation-based framework was the enabling technology to meet the research objectives. The framework enabled the highest fidelity analysis of Monte Carlo aircraft encounter simulations to date. It extends upon the development of probabilistic aircraft models and Monte Carlo aviation simulations. Generally only summary statistics are collected for each encounter with only a subset of individual encounters inspected with higher fidelity. The developed framework enables order

of magnitude greater amount of data to be processed, leading to the calculation of state-transitions for a MDP formulations. The flexibility of the framework did not constraint state selection nor the time horizon, enabling a high fidelity exploration of the aircraft encounter space. Observing all potential states from a Monte Carlo simulation instead of calculating each state during optimization, enables a new flexibility algorithmic design capability. Existing aircraft encounter models are limited to modeling aircraft at one second intervals, which can make it difficult how evolves over time. Analysis of Monte Carlo simulations based on these models overcame this limitation and lead to a new perspective on horizontal and vertical maneuvers.

In addition to exploring different states, the framework was leveraged to the development of new information theoretic metrics. Information theory and the concept of entropy had sporadically been applied to aviation safety and was mostly applied for strategic problems. There was only a few instances of applying these concepts to a tactical aviation problem like collision avoidance. The development and successful use of the near mid-air collision (NMAC) entropy and NMAC horizon metrics demonstrated the viability and potential of applying information theoretics to aviation safety problems. Specific contributions were made in quantifying risk and assessing an MDP prior to optimization and policy assessment.

ACAS X has been a very successful research and development effort that is paving the way forward for the next generation of operational aircraft avoidance. The initial clear focused towards manned aircraft, lead to some design decisions to limit the flexibility of the developed framework. Specifically, the barrier of the curse of dimensionality and the challenges to developing a horizontal and vertical action policy. The contributed simulation-based approach leverages the concept of MDP DP while introducing new flexibility for algorithmic design. This can easily be extended to different user classes such as rotorcraft and airships.

Finally, a new and alternative framework for aircraft MDP algorithm development was created. Leveraging this framework resulted in an extending collision avoidance algorithms to consider both horizontal and vertical maneuvers. An ACAS X framework can now be built to model and optimize this MDP formulation. Introducing additional optimization and MDP parameters to meet operational requirements and specific airspace modes can now begin. The simulation-based methodology however should result in less of these parameters, specifically with regards to response delay and maneuver coordination.

1.4 Outline

This document details the work completed over ten months associated with a M.S. thesis. Throughout this document, altitudes are in ft, vertical rates in ft/sec , airspeed in ft/sec . Time is reported in seconds (s). These units slightly differ from standard units used in aviation. The document consists of six chapters:

Chapter 1 introduces the thesis and it’s motivations, objectives, and contributions.

Chapter 2 overviews MDP optimization, aircraft avoidance, and encounter models.

Chapter 3 considers the formulation of the MDP and simulation-based framework.

Chapter 4 discusses how the framework from Chapter 3 was implemented.

Chapter 5 examines the formulations and the performance of the algorithms.

Chapter 6 summarizes this thesis’s contributions and future work.

Appendix A lists the states observed in the Monte Carlo simulations.

Appendix B defines different types of aircraft encounters.

Chapter 2

Background

This chapter provides relevant background material. First, it overviews MDPs and how they are used in optimization. Computational concerns for optimization follow. Next, a history of aircraft collision avoidance systems is provided with an emphasis on specific system. Finally, probabilistic models of aircraft encounters used in simulations, optimization, and safety studies are discussed.

2.1 Markov Decision Process Optimization

MDPs have been a topic of research since the 1950's and have modeled a wide variety of problems (Bellman, 1956; Bertsekas, 2005; Puterman, 2009), including aircraft encounters. The state of the world is assumed to evolve according to a fixed, potentially nonlinear, dynamic model. Solving an MDP involves searching for a strategy for choosing actions, also called a policy, that maximizes a performance metric, a cost associated with each combination of an action, current state, and future state.

A MDP's formal definition is a discrete time stochastic control process where there is some probability to transition from state s to s' where S is the a set of all states. Given a set of actions A , the transition matrix T indicates the transition probability for each pair of states in S . The cost matrix g encodes an associated cost for each transition. The work described implements a finite discretization for all

states, however this is not required. Mathematically, this is represented as a 4-tuple:

$$| S | < \infty \quad (2.1)$$

$$| A | < \infty \quad (2.2)$$

$$T_a(s, s') = P(s_{t+1} = s' | s_t = s, a_t = a) \quad (2.3)$$

$$g_a(s, s') = c_{(s_{t+1}=s' | s_t=s, a_t=a)} \quad (2.4)$$

The state-transitions are assumed to be Markovian, with sufficient information provided for the states. To be Markovian, the state-transitions must be “memoryless,” defined by Equation 2.5, where s_{t+1} only depends on s_t and given a present state s_t , no past states s_{t-1}, \dots, s_0 give no additional information about the future state s_{t_1} (Papoulis and Pillai, 2002). Previous work by Kochenderfer has validated the Markovian assumption for aircraft avoidance optimization problems (Kochenderfer and Chryssanthacopoulos, 2011).

$$P(s(t_n) \leq s_n(t) | s(t)) = P(s(t_n) \leq x_n | s(t_{n-1})) \quad \forall t \leq t_{n-1} \quad (2.5)$$

While a general MDP can be used when the state information vector is noisy, a partially observed Markov decision process (POMDP) is more suitable. A POMDP includes an additional observation model that generates an information vector based on the states. Several different POMDP solution strategies assume full observability using a DP algorithm (Littman et al., 1995; Hauskrecht, 2000; Fernández et al., 2006; Ross et al., 2008). A POMDPs formulation has been applied to aircraft avoidance problems (Kochenderfer and Chryssanthacopoulos, 2011; Bai et al., 2012). However the research discussed only considers a MDP formulation but can easily be extended to an POMDP.

2.1.1 Solving MDPs

While, MDPs can be solved in multiple ways, an DP based framework has shown to be effective for aircraft encounter problems. The optimal policy is defined by the optimal expected cost function J_K which is calculating via an iterative process. First, the function is initialized to zero for all states. Next, iterations of J_k are calculated given the function J_{k-1} until the desired horizon K .

$$J_0(s) = 0 \quad \forall s \quad (2.6)$$

$$J_k(s) = \min_a \left[g(s, a) + \sum_{s'} T(s, a, s') J_{k-1}(s') \right] \quad \forall 0 < k < K \quad (2.7)$$

$$J_K(s, a) = \min_a \left[g(s, a) + \sum_{s'} T(s, a, s') J_{K-1}(s') \right] \quad \forall k < K \quad (2.8)$$

The optimal K -step policy μ is the minimum of the expected cost function for all states and actions at each time step. Equation 2.9 defines the subproblem nature of DP, where the many smaller subproblems are solved and then combined to determine the overall optimal solution (Bellman, 1956).

$$\mu_K^*(s) = \arg \min_a J_K(s, a) \quad (2.9)$$

In practice, aircraft avoidance MDP problems have been solved where the state transitions and costs are calculated during optimization. The state-transition matrix probabilities and cost criteria are then provided as input to the optimization algorithm. This results in a tight coupling between the inputs and the potential states and difficult to add new states or controls. A simulation-based approach where the MDP is calculated prior to optimization has not been seriously considered prior. This can be attributed that is unfeasible, nor practical, to calculate a one-step transition matrix with large number of state variables across many dimensions for DP.

As of most physical systems, aircraft operate as continuous variables in NAS but are often discretized for simulation. The field of approximate DP has studied several different approaches (Powell, 2007) to address this, including the use of neural networks (Bertsekas and Tsitsiklis, 1995) and decision trees (Munos and Moore, 2002). A grid-based discretization (Davies, 1997) has shown to be a sufficient approximation for aircraft avoidance MDPs (Kochenderfer and Chryssanthacopoulos, 2011).

2.1.2 Shape Optimization

MDP DP is a specific approach to calculating an optimal policy but there are many ways to model and define the optimization itself. One general technique is simulation-based learning and models. In this regard, the simulation model can be thought of as a “mechanism that turns input parameters into output performance measures (Law et al., 1991).” The simulation itself is a function, whose explicit form is unknown, given a set of inputs and is represented as a set of numeric values (April et al., 2003). Given a function, optimization is very similar to the traditional iterative techniques. Defining the function is often a challenge but can significantly impact the optimization behavior, “Since simulations are computationally expensive, the optimization process would be able to search the solution space more extensively if it were able to quickly eliminate from consideration low-quality solutions, where quality is based on the performance measure being optimized (Laguna and Martí, 2002).”

A specific application of simulation-based learning is shape and topology optimization. Shape optimization is the process of optimizing the MDP formulation prior to calculating the optimal policy. A shape is considered optimal if it minimizes a certain cost function while satisfying known constraints. This is typically labeled as a state-space search (Torczon, 1997) and generally solved via numerical iteration.

Shape optimization can also be applied to the cost matrix. In principle, cost shaping is encoding additional information into the cost matrix in attempt to manip-

ulate the optimal policy. Conceptually, this can be viewed as giving the optimization algorithm a “hint.” It is considered a form of approximate DP (Powell, 2007) and have been applied to linear programming problems (de Farias and Van Roy, 2006) and selection of an appropriate cost matrix (de Farias and Weber, 2008).

Another technique are online costs, which are used by ACAS X (Asmar and Kochenderfer, 2013), the aircraft avoidance algorithm described in Section 2.3.3. This approach first generates an optimal policy using a fixed cost matrix. During policy execution, additional online costs are added and the expected cost for each state-transition and actions are calculated at runtime. An advantage is that the initial optimization does not include the state variables associated with the online costs, thus saving memory and computational power. However, a disadvantage is that the true optimal policy can only be identified via simulation.

A similar concept found in statistics is response surface methodology (RSM), which explores the relationships between explanatory and response variables (Box and Wilson, 1951). Through iteration, an easy to estimate and apply approximation of the optimal response can be obtained, regardless of the observed system. RSM starts with exploring which explanatory variables that influence the response variables. Once identified, these variables are used to build a more complicated model. Some extensions of response surface methodology deal with the multiple response problem (Myers et al., 2009).

Closely related to RSM is surrogate modeling which models the response and views the inputs as a “black box (Audet et al., 2000).” ACAS X has used surrogate modeling for specific operations, such as closely spaced parallel operations (Smith, 2013). As a form of shape optimization, it explores the design space and enables better prediction of the functional response (Jones et al., 1998). It is considered a better process to produce ideal ACAS X alerting behavior than hand tuning parameters.

2.2 Computational Considerations

Any MDP and optimization parameters must be stored in memory, for complex problems, such as aircraft encounters, memory requirements can become very large. The computational requirements for representing an MDP can increase due to a variety of factors such as expanding the state space or increasing the discretization of existing states. The computation efficiency and performance are important in evaluating the feasibility of the optimization.

A single simulated aircraft encounter can include over 50 states and metrics for each one-second interval. For hundreds of thousands individual encounters that can each last for a few simulated minutes, the computational requirements become very steep. Instead of storing all the simulation state outputs, Monte Carlo aircraft simulation results often include summary statistics supplemented with more in depth analysis of specific examples. Previous research have not compiled a complete distribution of simulation variables. Working with extremely large data sets are common to both commercial and research projects and significantly work have gone into developing systems to accommodate these data sets.

2.2.1 Storage and Access

Databases are used to store Monte Carlo aircraft simulation results. Parallel databases are designed to quickly ingest large datasets and provide reasonable access to (Dewitt and Gray, 1992) Google pioneered the technology with it's BigTable (Chang et al., 2008) technology, followed by Apache Accumulo (Apache Software Foundation, 2014), a column based "NoSQL" database (Stonebraker, 2010). Data is represented as a triple store of strings consisting of a row key, a column key, and a value that correspond to the entries of a sparse matrix. An advantage to triple stores over a relational schema, is that triple stores are optimized for storage and retrieval.

Accumulo and “NoSQL” provide Basic Availability, Soft-state, Eventual consistency (BASE) (Cattell, 2011), and guarantee that queries will provide the same answers eventually. This relaxed consistency contrasts traditional databases with a high level of Atomicity, Consistency, Isolation, Durability (ACID) (Cattell, 2011). High ACID databases guarantee that separate queries of the same data at the same time will give the same answer. A key advantage of a relaxed consistency database is that it can be built simply and provide high performance on commodity computing hardware. Accumulo is considered the one of highest performing databases and is widely used for government applications (Byun et al., 2012).

Leveraging the tuple and triple-store concepts, Dynamic Distributed Dimensional Data Model (D4M) (Kepner et al., 2013), a uniform mathematical framework based on associative arrays was developed. An associative array is an abstract data type that is a collection of keys associated with a value (Mehlhorn and Sanders, 2008). D4M doesn’t require a priori knowledge of the data for ingestion or parsing, so little a priori query optimization is required.

Accumulo and D4M have four specific features which make it ideal for recording complete Monte Carlo aircraft encounter simulations: row store, sparsity, unlimited columns, and high performance. Accumulo is a row store, so any row key can be looked up in constant time. The D4M schema stores both the database and its transpose, so any row or column can be looked up in constant time. This is due to data represented as extremely sparse tables, only non-empty columns are stored in a row. Additionally, there is no penalty for adding columns; resulting in the ability for unlimited columns. Finally, these features are enhanced by the high performance nature of Accumulo and D4M. Since it is parallel and distributed, many processes can modify the database across various tables in an efficient matter.

2.2.2 Matrix Representation

While databases store information, how to represent mathematically is also important, specifically MDP state-transition and reward matrices as square matrices. The basic data structure for a matrix is a two-dimensional array. Each entry in the array can be accessed by the two indices i (row) and j (column). A sparse matrix is a $m \times n$ matrix populated primarily with zeros as elements of the table (Pissanetzky, 1984). By contrast, if a larger number of elements differ from zero, then it is common to refer to the matrix as a dense matrix.

Sparsity corresponds to systems which are loosely coupled and is useful in combinatorics or physical dynamic models which have a low density of significant data or connections. Dynamics of physical objects, such as aircraft, are assumed to be loosely coupled due to kinematic constraints. For example, it is unrealistic for an aircraft to travel one mile in one second. This coupling can result in a diagonally dominant state-transition matrix. Equation 2.10 defines a diagonally dominant matrix where the magnitude of the diagonal entry in a row is larger than or equal to the sum of the magnitudes of all the other (non-diagonal) entries in that row. A symmetric diagonally dominant real matrix with nonnegative diagonal entries is positive semidefinite, an important optimization characteristic.

$$|P_{ii}| \geq \sum_{j \neq i} |P_{ij}| \quad \forall i \quad (2.10)$$

Special mathematical algorithms have been developed to leverage the structure of sparse matrices to improve computation speed (Yuster and Zwick, 2005), as quantified by Big O notation (Black, 2014). For dense matrices, the simplest algorithm for matrix multiplication of one $n \times m$ matrix and one $m \times p$ matrix is $O(nmp)$, thus two $n \times n$ matrices has complexity of $O(n^3)$. Sparse matrices are typically solved in parallel. If n is the number of non-zeros in the row, then the depth of the computation

is the depth of the sum, which is $O(\log n)$, and the work is the sum of the work across the elements, which is $O(n)$.

For an $m \times n$ dense matrix, enough memory to store up to $m \times n$ entries to represent the matrix is needed. Conversely, a sparse matrix only stores the nonzero elements and their indices and often eliminate operates on zero elements, leading to potentially substantial reduction in memory requirements. The memory savings are illustrated by Figure 2-1, which plots the memory requirement for a MATLAB $10,000 \times 10,000$ square matrix as a function of percentage of nonzero elements. For a 100,000,000 element matrix, the full dense matrix requires a constant 800,000,000 Bytes (0.8 Gigabytes) of memory. The full and sparse matrices have the same memory requirements when 50% of the elements are nonzero.

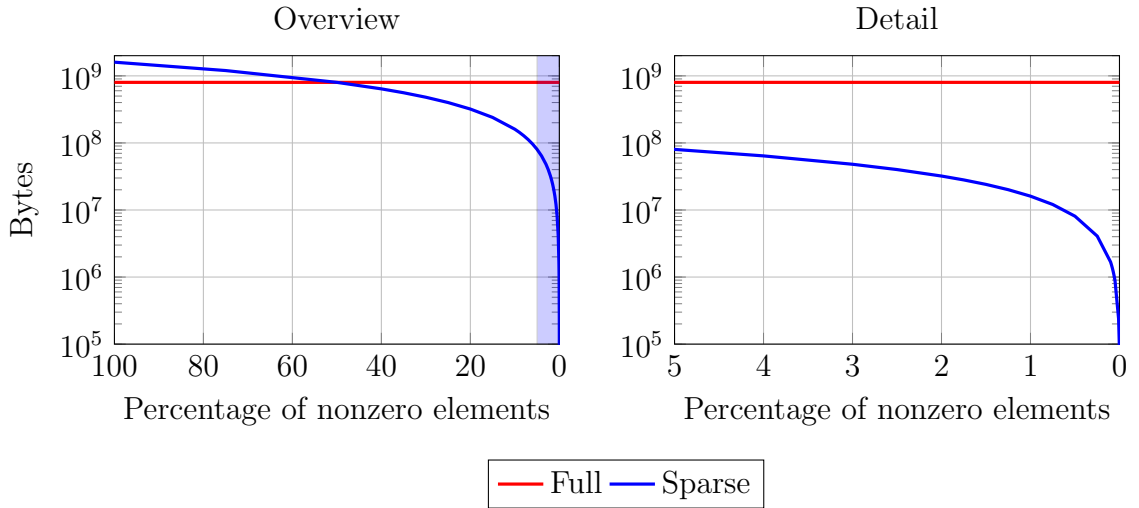


Figure 2-1: Memory required for MATLAB $10,000 \times 10,000$ matrix.

As the percentage of nonzero elements approaches zero, exponential memory savings can be realized using a sparse representation. The constant memory requirement for full dense matrices prevents MATLAB from holding large, multi-dimensional MDP matrices in memory.

2.2.3 Entropy

Given an allotment of memory constrained by the matrix representation, the entropy, a probabilistic representation of information content, can be quantified. From a finite sample of states, Shannon's entropy for an event X is (Shannon, 2001) :

$$H(X) = \sum_i Pr(x_i) I(x_i) \quad (2.11)$$

$$= - \sum_i Pr(x_i) \log_b Pr(x_i) \quad (2.12)$$

A common logarithmic base b is 2. A key characteristic is additivity, which asserts the quantity of entropy is independent of different parts of the process. Thus, the total entropy of a system can be calculated from the individual entropies of each of its sub-systems if the interactions between the sub-systems are known. Given a known interaction, then the conditional entropy of two events $X|Y$ can be calculated:

$$H(X|Y) = \sum_{i,j} p(x_i, y_j) \log \frac{p(y_j)}{p(x_i, x_j)} \quad (2.13)$$

Entropy is a seminal metric in communications (Fang et al., 1997) and is widely used as weighting for optimization (Bejan, 1995) or state-space exploration (Doye and Wales, 1998). However, it has sparsely been applied to aviation safety systems. Entropy has represented the total population of aircraft operating in an airspace to guide an strategic air traffic optimization method (Lv et al., 2013); focusing on minimizing the occurrence of aircraft encounters, rather than issuing collision avoidance maneuvers. Another strategic effort used defined an aircraft obstacle avoidance problem where entropy quantified terrain (Doebbler et al., 2005).

For close-encounters, Šišlák proposed an autonomous aircraft avoidance algorithm using a game theory approach where entropy defines the amount of information provided to each aircraft in a Nash equilibrium (Šišlák et al., 2009). This is analogous

to defining entropy to quantify coordination between aircraft or probabilistic knowledge of the intruder’s aircraft potential maneuvers. Lai uses a relative entropy rate for target tracking and acquisition for a vision-based UAS sense and avoid (SAA) system (Lai, 2010). Entropy concepts have been used to calculate aggregate weights for evaluating airborne weapon systems (Mon et al., 1994; Cheng, 1997), a similar problem where the objective is to collide rather than avoid.

2.3 Aircraft Avoidance Algorithms

The previous sections established the theoretical and computational aspects of MDP and simulation-based learning. The primary research objective is to develop an aircraft avoidance algorithm. Understanding current and future manned and UAS avoidance algorithms are important to developing design criteria and decisions.

Regardless for use on manned or unmanned aircraft, these systems need to prevent or minimize the risk of a NMAC. A NMAC is often used a surrogate for collisions when evaluating safety performance. It is defined as an incident associated with the operation of an aircraft in which a possibility of collision occurs as a result of proximity of less than 500 ft horizontally and less than 100 ft vertically to another aircraft, or a report is received from a pilot or a flight crew member stating that a collision hazard existed between two or more aircraft. An aircraft avoidance maneuver algorithm will issue a maneuver advisory, a recommended maneuver to reduce NMAC risk. The algorithm needs to consider a variety of factors, any of which can could be potential state variables in an optimization:

- Aircrafts’ performance limits
- Aircrafts’ position and airspeed
- Aircrafts’ dynamic rates
- Airspace class
- Effect on air traffic control (ATC)
- Number of aircraft

- Operational suitability
- Pilot or aircraft response
- Right of way rules
- Traffic density

As technology and the NAS evolves, different collision avoidance capabilities and systems have been developed. Aircraft collision avoidance has been identified as a global safety requirement since a series of mid-air collisions in the 1950s.

The 1956 collision between two airliners over the Grand Canyon, the first commercial airline crash to result in more than 100 deaths, highlighted the collision risk and led to the establishment of the FAA in 1958. These collisions spurred both the airline and aviation authorities to initiate development of effective collision avoidance systems (Schlager, 1994). Morrel provided the first mathematical representation of the aircraft avoidance problem (Morrel, 1956) and subsequent early attempts of collision protection include strategic airspace design and tactical ATC.

For several decades thereafter, a variety of approaches to collision avoidance were explored, until 1974, FAA narrowed its focus to the Beacon Collision Avoidance System (BCAS), a transponder-based airborne system that was operationally suitable only in low-density airspace. In 1978, a mid-air collision near San Diego resulting in 144 fatalities led to the expansion of the collision avoidance effort; in 1981, the name was changed to the TCAS and developed for all airspaces (Harman, 1989). TCAS is now required worldwide for manned aircraft ¹ and provides a collision avoidance functionality as one part of a layered air traffic management (ATM) architecture.

With the advent of UAS integration into the NAS, the integrated ATM system is comprised of independent conflict management layers to mitigate collision risk with manned and unmanned aircraft taking different responsibilities. UASs require a unique SAA capability to maintain well clear from and avoid collisions with other

¹Aircraft with a maximum take-off mass of over 5700 kg (12,600 lb) or authorized to carry more than 19 passengers.

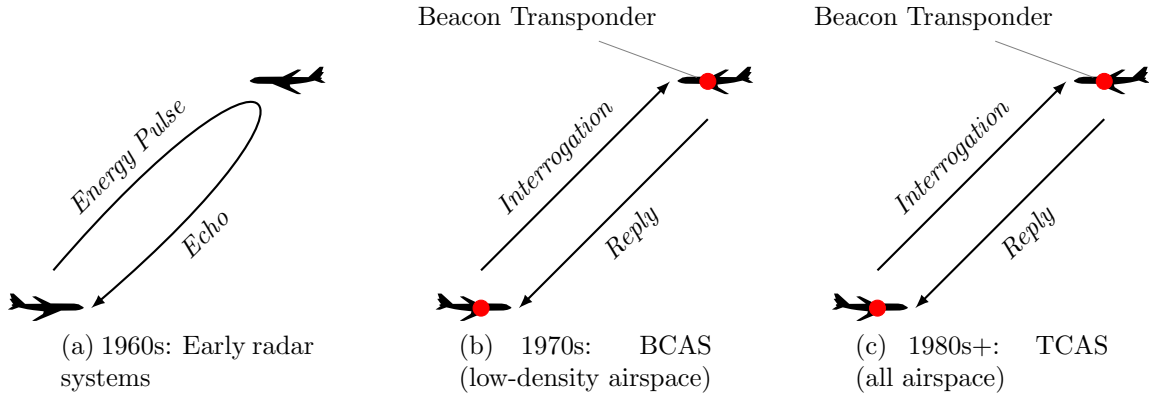


Figure 2-2: Development and evolution of airborne collision avoidance

airborne traffic (Federal Aviation Administration, 2013). The FAA has not (nor will ever) certify the use of TCAS as an acceptable alternative for UAS SAA requirements. A supporting argument for this lack of certification is that historical manned aircraft performance assumptions are not applicable to UASs. In response, new collision avoidance algorithms are being developed along with UAS-specific systems that also must meet a self separation requirement to maintain well clear of other aircraft (Code of Federal Regulations, 2011).

Other ATM system evolutions, such as future airspace requirements and implementing of NextGen ATM, requires a major overhaul of TCAS and collision avoidance philosophy. New collision avoidance systems need to function with a variety of different possible surveillance sources, such as automatic dependent surveillance-broadcast (ADS-B) (Lacher et al., 2007). For UAS, collision avoidance systems should be cognizant of self separation systems that protect against well-clear violations. Traditionally it is viewed that self separation leverages gentler horizontal maneuvers while collision avoidance implements stronger vertical maneuvers leading to a layered SAA architecture.

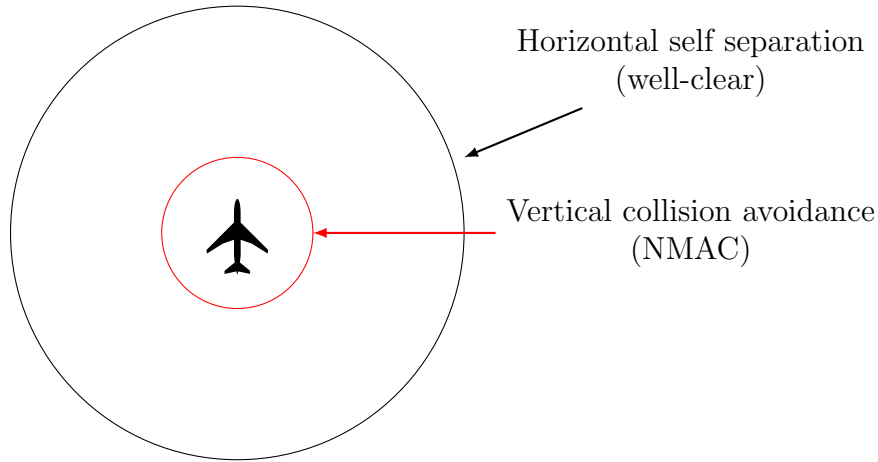


Figure 2.3: Traditional collision avoidance and self separation layered architecture

2.3.1 TCAS

The TCAS was the first and only airborne collision avoidance system to be mandated on all large manned transport aircraft. It uses an on-board beacon radar to monitor the local air traffic and logic to determine when to alert pilots to potential conflicts (Williamson and Spencer, 1989) and has been shown to significantly improve safety (Espindle et al., 2009; European Organisation for the Safety of Air Navigation, 2011). TCAS consists of a sensor and algorithm; the algorithm is implemented as pseudocode designed through human expert opinion. The coupling of the sensor to the algorithm and the pseudocode design makes it very difficult to modify or upgrade TCAS.

TCAS implements a five dimension state space and does not directly consider any horizontal states. This is a consequence of the TCAS sensor which provides significantly better aircraft estimates in the vertical axis than horizontal. Additionally, due to the reliance on beacon surveillance, TCAS does not protect against aircraft that are operating without a transponder. Thus the TCAS state space consists of the following and is illustrated by Figure 2.4:

- τ_h : TCA horizontally,
- \dot{h}_i : vertical rate of the intruder, and
- Δh : relative altitude,
- \dot{h}_o : vertical rate of ownship
- s_{RA} : state of RA.

These states were not selected via an analytic approach due to the technology available in the 1970's and 1980's. Instead states that could be easily correlated with observed aircraft dynamics (i.e. vertical rate) or the output of the surveillance system were selected. However some assumption are no longer appropriate. For example, level flight was defined as a vertical rate of ± 10 ft/s (600 ft/min) (Clark and McFarland, 1977), yet some UASs' maximum vertical rates are less than ± 10 ft/s.

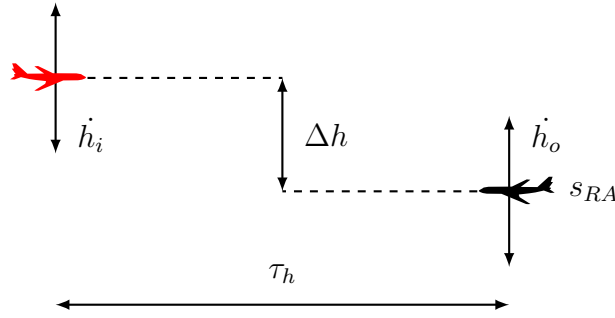


Figure 2.4: TCAS states

The TCAS algorithm mimics many of the concepts put forth by Morrell in 1956 (Morrell, 1956). The τ aggregate feature was introduced as a sufficient approximation of separation due to either poor surveillance or to simplify the dynamic model due to the complexity of calculating the separation distance. τ can be considered a countdown to collision and for unaccelerated flight the true time till collision and τ are sufficiently similar. The TCAS algorithm defines a close encounter as potential collision between aircraft within $\tau_h \leq 45$ s, which is the range of the 30–60 s minimum alert (“escape”) time originally recommended by Morrell. Finally, three states quantify the vertical axis and s_{RA} is a discrete variable that represents the advisory.

An advisory is either a traffic alert (TA) or RA. A TA notifies pilots that another aircraft is in the vicinity. A RA is a stronger advisory when there is significant NMAC risk and is a direct vertical maneuver to prevent a conflict. The TA and RA alerting regions can vary by altitude layer, as illustrated by Figure 2-5.

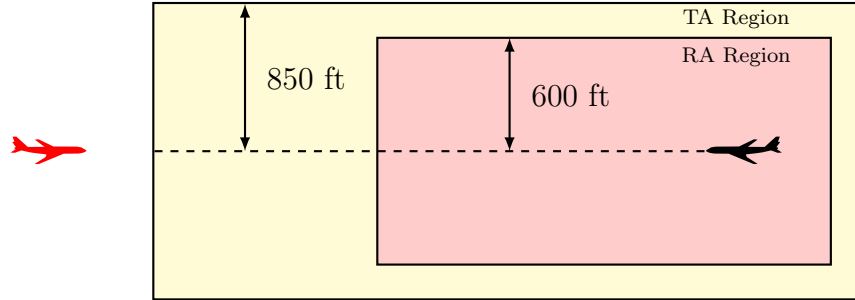


Figure 2-5: Example TCAS altitude criterion for 5000–10,000 ft.

Generally, a TCAS RA is either preventive or corrective and issues an RA 40 s or less prior to a safety conflict. A preventive RA, such as maintain or do not climb, is a maneuver that prevents a riskier encounter geometry and is generally a less disruptive maneuver. Table 2.1 lists all TCAS corrective RAs actions.

Table 2.1: TCAS correction RA action set.

Command	Description
COC	Clear of conflict
MLTO	Multithreat level off or composite advisory
MDES	Maintain descend
MCL	Maintain climb
DES2500	Descend at 2500 ft / min
CL2500	Climb at 2500 ft / min
DES1500	Descend at 1500 ft / min
CL1500	Climb at 1500 ft / min
DNC	Do not climb
DND	Do not descend

During initial development, TCAS implemented both horizontal and vertical maneuvers (Clark and McFarland, 1977; Zeitlin, 1979). Due to the simpler dynamics of the vertical axis and poor azimuth accuracy of the TCAS sensor, horizontal maneuvers were removed as TCAS matured from a research to operational system. However,

the TCAS architecture allowed for future research of horizontal maneuvers, as evident by left-over horizontal maneuver code in the TCAS pseudocode.

2.3.2 Horizontal TCAS

During initial TCAS development, horizontal maneuvers were considered “passive” while vertical maneuvers as “active” and designed so that a combination of horizontal and vertical maneuvers could be displayed to the pilot. After TCAS simplified the algorithm to only leverage vertical maneuvers, subsequent efforts to incorporate horizontal maneuvers include Enhanced TCAS II and TCAS III.

Enhanced TCAS II was developed by the Bendix Corporation in the first half of the 1980’s. It implemented only horizontal maneuvers with the algorithm considering horizontal acceleration and miss distance error (Sinsky et al., 1984). Additional sensors and surveillance information were incorporated to address the poor azimuthal accuracy of the TCAS sensor. Specifically, the algorithm uses the standard deviation of the horizontal miss estimate, σ_D and σ_B , the angular rate’s standard deviation:

$$\sigma_D = (v_o + v_i)\tau_h^2\sigma_B. \quad (2.14)$$

The algorithm implements a straight line linear projection and attempts to achieve a horizontal miss distance approximately a third of the potential horizontal displacement, as depicted by Figure 2.6. The algorithm designer’s claimed that a safe horizontal maneuver is possible if the error in estimating miss distance is smaller than the aircraft’s ability to outmaneuver that error.

In the second half of the 1980’s MIT Lincoln Laboratory (MIT LL) explored the feasibility of a probabilistic method for selecting between horizontal and vertical maneuvers (Wood, 1987). They explored the dynamics of aircraft avoidance maneuvers and associated operational impacts. Due to NMAC defined as greater in the horizon-

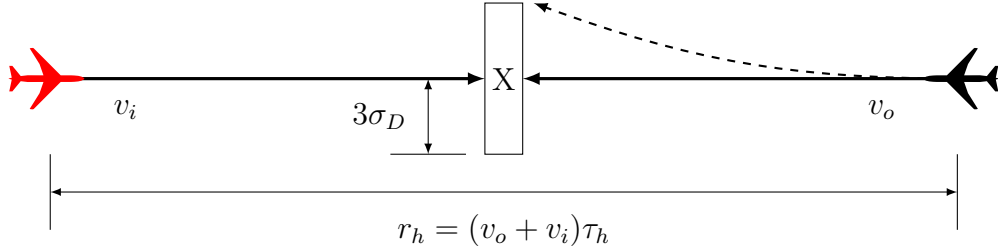


Figure 2.6: Enhanced TCAS II algorithm (Sinsky et al., 1984).

tal plane (500 ft) than the vertical axis (100 ft), it simply takes longer to maneuver and requires greater displacements. For commercial manned aircraft where vertical maneuvers were shown to be sufficient, the benefits of including horizontal maneuvers could be minimal. It is noteworthy because this probabilistic model did not leverage τ , instead directly using horizontal and vertical separation at closests point of approach (CPA) and true aircraft states.

Majority of horizontal TCAS research and development focused on surveillance and bearing measurements rather than robust algorithm designs. Achieving an acceptable position and bearing rate error were not feasible during development (Burgess et al., 1994; Burgess and Altman, 1995) and is cited as a major factor in not transitioning these algorithms to an operational capability. With recent advantages in surveillance technology, sufficient azimuth measurements can now be obtained.

2.3.3 ACAS X

In 2009, the FAA TCAS Program Office began formal research on the next generation of collision avoidance system designed to improve upon the level of safety and operational performance provided by TCAS, termed ACAS X (Holland et al., 2013). It adopts a completely different design methodology than TCAS based on decision theory (Kochenderfer et al., 2012). This new approach involves automatically deriving the optimal logic based on explicit probabilistic models and cost functions that represent the objectives of the system. Development is then focused on choosing

models and cost parameters to achieve safety and operational performance objectives rather than modifying pseudocode like TCAS. In addition to greatly simplifying the development and maintenance of the system, ACAS X accommodates a variety of different sensor systems, enabling new procedures and user classes. There are four ACAS X variants actively being developed (Holland et al., 2013):

- *ACAS Xa* (active) is intended to replace TCAS. It incorporates active transponder-based observations to provide global protection against tracked aircraft.
- *ACAS Xo* (operation) provides operation-specific alerting during procedures such as closely-spaced parallel runway operations. Xo facilitates procedure-optimized alerting against a user-selected aircraft while providing global Xa protection against all other traffic (Smith, 2013).
- *ACAS Xu* (unmanned) is designed for UASs and accepts a variety of surveillance inputs and uses logic optimized for a UAS performances (Brooker, 2013).
- *ACAS Xp* (passive) will be used on low-performance general aviation aircraft and helicopters that currently lack certified SAA capability. It passively receives surveillance information and provides vertical guidance optimized for the expected range of aircraft performance (Billingsley et al., 2012).

General Overview

ACAS X consists of three systems: surveillance, logic, and display. The surveillance system detects and tracks local air traffic and through a use of weight samples, represents an aircraft's state estimate. Unlike TCAS which is optimized for only the one transponder-based surveillance system, ACAS X is more flexible and configured as a plug-and-play system. This is a benefit to the decision theory architecture. Given that a surveillance system meets certain performance criteria, it can be integrated

into ACAS X; enabling support for ADS-B or non transponder-based surveillance systems, such as electro-optical and infrared sensors.

The logic system uses the weighted state samples as input and decides which advisory, if any, to display to the user. The ACAS X logic is tuned using a structured, iterative, computational DP process to improve relative to operational and safety goals. The ACAS X cost function incorporates factors pertaining to safety and operational performance. For example, a state which is associated with an NMAC will incur a penalty cost to discourage transitions into an NMAC-related state. Through using MDPs, the logic accounts for uncertainty in the aircraft dynamics and pilot response, leading to significantly improved robustness compared to TCAS (Chryssanthacopoulos and Kochenderfer, 2011), while handling multiple simultaneous threats (Chryssanthacopoulos and Kochenderfer, 2012).

The display system provides the user with the recommended advisory. The display system is dependent upon the ACAS X variant and user class. Unlike TCAS, ACAS X does not assume users are a trained private pilot (McCarley and Wickens, 2004). The display must also be accessible enough to maximize pilot compliance (Pritchett et al., 2012). As ACAS X transitions to operational capabilities, display research will become a greater focus.

Finally, the common ACAS X framework results in these variants being interoperable with each other and with legacy versions of TCAS. For variants with similar controls, ACAS X coordinates complementary maneuvers with either ACAS X or TCAS over a datalink (Kochenderfer and Chryssanthacopoulos, 2011). This coordination is achieved by imposing constraints in the cost function.

Logic Design

Of the variants, ACAS Xa is the most mature and ACAS Xu is one of few attempts to fully develop a horizontal action algorithm in over a decade. ACAS Xa was the

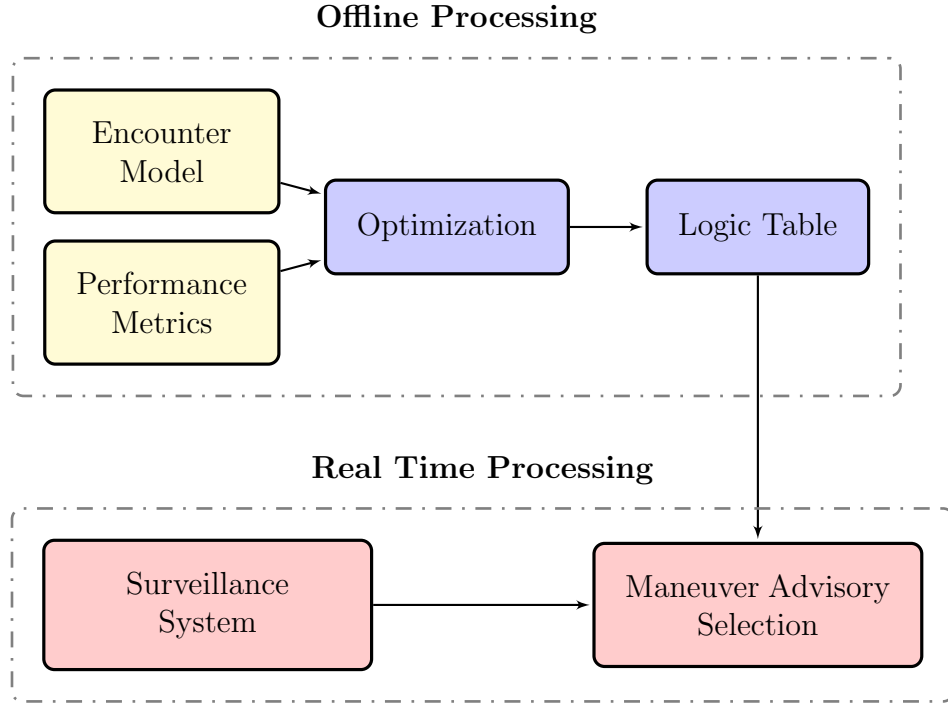


Figure 2·7: ACAS X logic development approach. Nodes in yellow represent the models, blue nodes represent the computer optimization process based on the model, and those in red represent the implementation at runtime.

first variant developed and provides both the theoretical foundation and code base for all other variants. A fifteen month tuning effort to optimize costs and focused on improving performance using existing transponder-based surveillance culminated in a proof-of-concept flight test in 2013 (Holland et al., 2013). The multi-year tuning effort highlights the difficulty in generating the cost matrix.

Unlike TCAS which leverages a deterministic pilot response model and linear extrapolation to predict future states, ACAS X implements a probabilistic pilot response. It has been observed via radar data that there is significant variability significant variability in the delay and strength of the response of pilots to advisories (Kuchar and Drumm, 2007). This is reasonable since humans are almost never deterministic in anything (Rasmussen, 1983).

ACAS Xa shares the same action command set as TCAS, previously described in Table 2.1 and state state described in Section 2.3.1. This is rational because ACAS Xa will replace TCAS. Instead of psuedocode, ACAS Xa is the implementation of an optimal MDP using a state space represented as a multidimensional discretized grid. The proof-of-concept DP algorithm used horizontal range instead of τ_h and the model parameters were chosen by hand without any analytical assessment (Temizer et al., 2009). As the approach evolved into ACAS X program, majority of the research effort focused on the optimization process and parameters, rather than the state-transition representation. These initial efforts recognized that resources such as the probabilistic encounter models exist and could for used for state space exploration. This reorganization was a contributing factor to the development of the research objectives described in Section 1.2.

ACAS Xu arose from the limited vertical capabilities of many UASs and the development of self separation SAA. Some UAS can only acheive vertical rates of approximately ± 12.5 ft/sec (750 ft/min), half of what TCAS and ACAS Xa assume for the vertical maneuvers. With the rapidly growing need for UAS airspace access, a comprehensive state space search was not feasible. Instead, ACAS Xu adopted a similar approach to the ACAS Xa by reducing an axis into a τ state; whereas ACAS Xa reduced the horizontal plane to τ_h , ACAS Xu reduced the vertical axis into a vertical aggregate feature τ_v . The horizontal plane is then decomposed into first order states, resulting in a seven dimension state space:

- r_h : horizontal range,
- ψ_i : own aircraft heading,
- $\dot{\psi}$: relative heading rate,
- v_o : own aircraft airspeed,
- v_i : intruder aircraft airspeed,
- τ_v : time of closet approach (TCA) vertically, and
- s_{RA} : state of the RA.

The state space is larger due since the horizontal axis is a plane whereas the vertical axis is geometric line. Relative geometry has four quadrants, unlike ACAS Xa where relative geometry only has two (above and below).

ACAS Xu does not currently implement a uniform grid discretization. This is due to a variety reasons but foremost is that the horizontal spatial space is spatially larger. Whereas the maximum separation in ACAS Xa is ± 4000 ft vertically, ACAS Xu has a maximum separation of 60,760,ft (10,miles) horizontally. ACAS Xu must also account for π wrap around for the heading variables which significantly increases the number of states in these dimensions. The corresponding increased memory requirements prevent a simple merger of the ACAS Xa and ACAS Xu state spaces.

2.3.4 Non-Dynamic Programming Based

In addition to the upgrading and maintaining TCAS (Espindle et al., 2009) and the development of dynamic programming-based SAA via the ACAS X program, other SAA effort have taken place over the past decade. The Multiple Intruder Autonomous Avoidance (MIAA) program leveraging the Jointly Optimal Collision Avoidance (JOCA) algorithm (Graham et al., 2011) and the U.S Navy’s Triton program implementing due-regard operations (Lutz et al., 2013) are briefly discussed with emphasis on how they represent the aircraft avoidance problem.

Jointly Optimal Collision Avoidance

The JOCA algorithm is part of the MIAA system currently in development (Graham et al., 2011) whose goal is to develop a UAS capability to autonomously detect and perform collision avoidance against all aircraft classes. MIAA has the capability to ingest position and velocity information; ownship intent information such as current control values and way points; and TCAS information.

JOCA provides the optimal trajectory control solution to the Multi-sensor Inte-

grated Conflict Avoidance (MuSICA) system (Graham et al., 2011). JOCA is a type of online model predictive control (MPC) algorithm that selects a maneuver based on minimizing a cost function at each time step and uses an internal aircraft response model to predict the trajectory of the unmanned aircraft. Figure 2-8 shows the maneuver selection hierarchy used by JOCA’s MPC algorithm. Instead of optimizing over a state space, JOCA simply calculates each potential trajectory and is limited by the computational power required to calculate a set of trajectories. Additionally, for intruders with a transponders, JOCA requires TCAS.

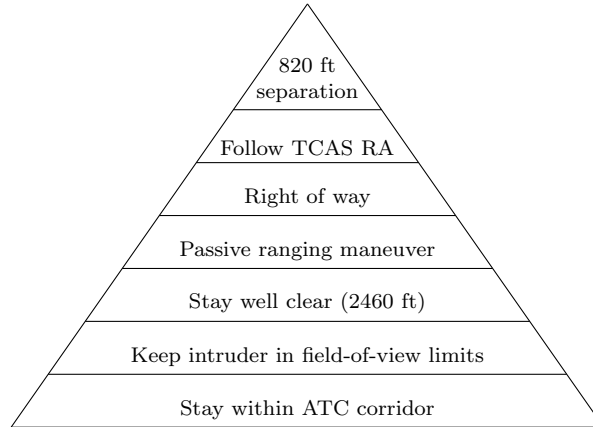


Figure 2-8: JOCA MPC maneuver selection hierarchy.

It takes course deviation into account by calculating how much the maneuver will deviate from its intended course. It also estimates how a maneuver will affect its estimate of the intruder’s position by considering where the intruder will be in the sensor field of regard and using an extended Kalman filter tracker (Chen et al., 2011). If the intruder is noncooperative, the aircraft has to be able to continuously track the intruder aircraft with its on-board sensors. Passive ranging maneuvers can also increase the confidence of the estimate of the intruder’s relative position and velocity (Shakernia et al., 2005). It chooses maneuvers that, when appropriate, comply with right of way rules for aircraft (ICAO, 1990) and with TCAS RAs.

Due Regard

A different approach is to provide guidance to the UAS human controller, who will decide and issue SAA commands. As part of the due regard safety case, this approach relies on minimizing UAS operations in high traffic densities. Simulations to assess the potential efficacy of the approach leverages an advisory logic developed from pseudocode based on expected human responses, instead of a mathematical model like a MDP or MPC. The pseudocode is based on Department of Defense (DOD) Instruction 4540.01 which describe procedures for military aircraft to conduct operations in international airspace that cannot be performed under ICAO flight procedures. These options include (Department of Defense, 2007):

- Aircraft shall be operated in visual meteorological conditions.
- Aircraft shall be operated within surveillance and radio/satellite communications of a surface and/or airborne facility.
- Aircraft shall be operated outside controlled airspace.

From an optimization perspective, it is extremely difficult to represent these options as a discrete state space, many different discrete states would be required. The state space would rapidly expand and a comprehensive representation has not been shown to be feasible. As such this approach provides little insight into computationally efficient state spaces. Research efforts focus more on developing pilot models and quantifying human decision making abilities rather solving for the mathematically optimal maneuver.

2.4 Encounter Models

The ACAS X MDP approach leverages aircraft encounter models to help calculate state-transitions. A simple white-noise transition model is used for optimization and

a higher fidelity encounter model is leveraged to evaluate the ACAS X optimal policy. Aircraft encounters models were developed prior to the development of ACAS X and can be attributed as one of the primary factors why ACAS X has been successful.

An aircraft encounter model is a statistical model that mathematically represent how aircraft behave during close encounters. An encounter model must accurately represent the distribution of encounters where a SAA system would be likely to alert; otherwise, the output of the system evaluation will be erroneous. Specifically, an encounter model shall:

- Realistically capture the initial relative geometry of the aircraft;
- Be built using aircraft operational data;
- Reflect realistic aircraft flight dynamic
- Have a representative time period for different evaluations;
- Have dependencies on geographic region, airspace class, and altitude layer;
- Enable fast-time simulation;

2.4.1 Types and Use

Early encounter models were built to support the development and certification of TCAS (McLaughlin, 1997; MITRE, 1983). These models capture the behavior of aircraft in encounters that TCAS was expected to resolve—that is, encounter situations with cooperatively equipped aircraft. In the past decade, more advance models have been developed to evaluate the performance of SAA systems (Kochenderfer et al., 2010b). The four primary types are:

- **Uncorrelated Encounter Model of the National Airspace System:** used to evaluate the performance of SAA systems when at least one aircraft is nonco-

operative or neither aircraft is in contact with ATC (Kochenderfer et al., 2008b; Weinert et al., 2013).

- **Correlated Encounter Model of the National Airspace System:** used to evaluate the performance of SAA systems when both aircraft are cooperative and at least one aircraft is receiving ATC services (Kochenderfer et al., 2008a).
- **Encounter Models for Unconventional Aircraft:** used to evaluate the performance of an SAA system when encountering unconventional aircraft, defined as aircraft unlikely to carry a transponder (Kochenderfer et al., 2009).
- **Due Regard Encounter Model:** used to evaluate SAA systems for UAS flying due regard in oceanic airspace (Griffith et al., 2013).

Each of these encounter models includes variables that account for variations in encounters with respect to different airspaces. For example, one of the variables in the uncorrelated encounter model is Airspace Class, which includes B, C, D, and O (Other). Table 2.2 indicates the appropriate model to use based on the study being performed.² For the offshore environment, the correlated and uncorrelated encounter models encompass encounters more than 1 NM beyond the shore and the due regard model begins at 12 NM, where due regard flight is permitted.³ The oceanic environment includes international airspace beyond radar coverage. Note that no existing model covers encounters between two IFR aircraft in oceanic airspace. The reason for this is that one cannot observe encounters of sufficient fidelity in the data feeds. Similarly, there is no model that covers encounters with visual flight rules (VFR) or noncooperative aircraft in oceanic airspace due to a lack of surveillance data. If a collection of encounters for these types of encounters is required, they should

²Note that a model does not exist for combinations without a mark.

³Note that one cannot build a *correlated* encounter model from radar data for due regard flight in the offshore environment because one does not observe a sufficient number of encounters between instrument flight rules (IFR) and non-IFR traffic beyond 12 NM.

be built based on best assumptions about aircraft behavior and should leverage data from similar encounter models as is necessary. For example, one could use offshore models or enroute model of the Contiguous United States (CONUS).

Table 2.2: Encounter model categories.

Aircraft of Interest		Intruder Aircraft			
Location	Flight Rule	IFR	VFR	Noncooperative Conventional	Noncooperative Unconventional
CONUS	IFR	C	C	U	X
	VFR	C	U	U	X
Offshore	IFR	C	C	U	X
	VFR	C	U	U	X
	Due Regard	D	U	U	X
Oceanic	IFR				
	VFR				
	Due Regard	D			

2.4.2 Development and Structure

To develop the uncorrelated and correlated encounter models, radar data was used from the 84th Radar Evaluation Squadron (RADES) at Hill AFB, Utah. RADES receives radar data from FAA and Department of Defense sites throughout the United States. They maintain continuous real-time feeds from a network of sensors with radar ranges varying from 60 – 250 nm. Recently these models now includes the offshore environment out to the limits of radar coverage (Weinert et al., 2013). The unconventional aircraft encounter models were developed using weather balloon data and GPS data (Edwards, 2010) and the due regard model with self-reported positions of aircraft flying in oceanic airspace (Griffith et al., 2013).

The raw radar data are first processed using a tracking algorithm developed at

MIT LL (Grappel, 2001). A fusion algorithm, also developed at MIT LL (Gertz, 1983), then fuses tracks from multiple sensors. After the raw tracks have been developed, they are passed through an outlier detection algorithm, smoothed and interpolated. From the interpolated tracks, features are extracted to build the encounter models. A detailed explanation of processing the radar data into the encounter model can be found in the documentation of other models (Kochenderfer et al., 2010b). Recent processing advances including implementing outlier detection algorithms and rejection sampling to improve the accuracy of the models for longer simulation times (Weinert et al., 2013).

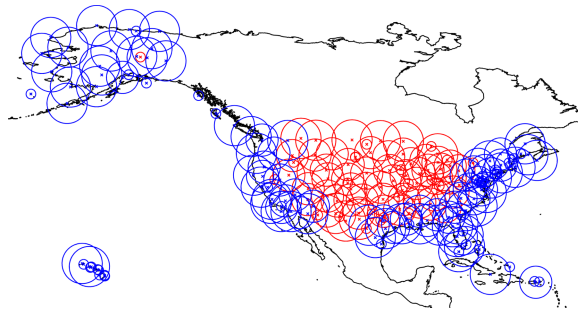


Figure 2.9: RADES sensor coverage map ignoring terrain masking. Radars in red theoretically only provide data over CONUS.

The current model structure is a Bayesian network, a representation of a multivariate probability distribution as a directed acyclic graph (Murphy, 2002). Each node in the network structure represents a variable and arrows represent conditional dependencies between variables. There are two Bayesian networks in an encounter model: an initial network to set up an encounter, and a transition network to describe how the variables specifying the trajectories evolve over time. The dynamic Bayesian network structure is learned by maximizing the posterior probability of the network structure given the data. Dynamic behavior is modeled in a dynamic Bayesian network as a MDP. Aircraft turn rate, airspeed acceleration and vertical rate may change once per second. Given a set of initial conditions and these dynamic variables, the

aircraft trajectories in the encounter can be constructed.

By sampling to produce trajectories many times, a Monte Carlo simulation defined by the encounter model is possible. By aggregating the results of the individual simulations, a distribution of an unknown probabilistic state is calculated (Metropolis and Ulam, 1949). Together the samples are representative the airspace with Monte Carlo simulations primarily supporting simulation analysis.

Optimization Use

Historically they have been analysis tools because they only contain a subset of potential aircraft avoidance state variables. While the encounter models are Markovian with an exhaustive state-transition matrix that specifies the probability of transitioning between all pairs of states. However, the number independent parameters required to define the matrix grows super-exponentially with the number of variables defining the model, as the “curse of dimensionality” affects optimization and Bayesian models alike. The more independent parameters there are in the model, the more data one needs to properly estimate their values. Leveraging dynamic Bayesian networks reduces the number of parameters but does not completely address the exponentially growth associated with adding new state variables.

Therefore for simulation-based optimization, the encounter models can not directly be used because they do not contain sufficient information for optimization. For example, the uncorrelated encounter model has no state variables that are associated with range between aircraft. Costs based on safety metrics, such as NMAC, can not be encoded. This leads to the requirement to simulate the aircraft encounter and record the required state variables. The simulation output, based on encounter model input, would address the research objectives; the development of this metamodel is discussed in detail in Chapter 3.

Chapter 3

Theory and Approach

The previous Chapter 2 overview aircraft avoidance efforts and MDP DP optimization. This chapter details the theory and approach used to meet the research objectives described in Section 1.2. The overall problem from the theoretical perspective for collision avoidance is first developed and leads into formulation of an aircraft avoidance MDP via simulations. For each of these MDP elements, a comprehensive assessment of potential states, including aggregate features, is provided. An NMAC risk entropy metric is then defined to assist in evaluating each potential state. Finally, the approach to evaluate the feasibility of the simulation-based MDP is discussed. The following Chapter 4 discusses the technical details of implementing these concepts.

3.1 Overview

For optimization, the abstract conceptual avoidance problem needs to be represented as a mathematical numeric system. As described in Section 2.1, a probabilistic MDP with states, actions, and costs is used. The MDP must quantify the following abstract questions:

- What is the frequency of each state-transition?
- What is the safety and operational risk for a given state-transition?

The optimization challenge is then selecting a set of states that sufficiently quantify both the safety risk while enabling transitions between discrete states. Traditionally

aircraft avoidance algorithms decompose either the horizontal plane or vertical axis into a τ state where $\tau = 0$ indicates a complete loss of separation in that axis. The algorithm can then select an action in the other axis to increase separation. TCAS and ACAS Xa decomposes the horizontal plane into τ_h and select vertical actions while ACAS Xu decomposes the vertical axis into τ_v and selects horizontal actions. Since a joint horizontal and vertical action space requires sufficient information about both the horizontal plane and vertical axis, neither can be decomposed into a τ state. This leads to adding additional states and the “curse of dimensionality.”

Either due to lack of computational power or historical success of using a τ state, there has been limited to no prior research on exploring and quantifying potential states for optimization. The success of the encounter models show that aircraft dynamics can be modeled via a probabilistic processes but they do not quantify the safety and operational risk. Nor were aggregated features considered during encounter model development. Specifically both TCAS and ACAS X selected states by hand. The ACAS X designers also recognized that there is a wealth of available data for intelligent state selection (Temizer et al., 2009) but never explored this space.

Therefore to assist in the development of a joint-action MDP formulation, an exploration and search over potential variables is required. Modifying the existing ACAS X MDP framework and dynamics for each new potential variable would be time consuming and inefficient. Instead a new simulation-based framework is implemented, leveraging recent advances in signals processing and databases. The framework can quickly and efficiently calculate the joint conditional state probabilities and is flexible enough to support any set of states. The entropy for each state and state-transition quantifies how memory efficiency and how well it quantifies NMAC risk. Applying shape optimization, as described in Section 2.1.2, could reduce the memory requirements of the state-space while persevering the utility of the space. The simulation

itself will be Monte Carlo aircraft encounters sampled from the encounter models. A MDP formulation will be built using this simulation-based approach to demonstrate the feasibility of the formulation. It will be optimized using an alternative, yet similar, approach to ACAS X.

The developed MDP assumes that decisions are made once per second, the same same frequency at which TCAS and ACAS X. Only single intruder encounters will be considered, as multi-intruder scenarios are relatively rare. However, through utility fusion multiple intruder aircraft can be considered in parallel without expanding the state space (Kochenderfer and Chryssanthacopoulos, 2011). Finally, ownship assumes that the intruder is not equipped with an SAA system.

This alternative approach is not designed to replace ACAS X but focuses on providing more flexibility in algorithm design. There are other approaches to reduce memory requirements. Due to the potential organized structure and combinatorial nature of discrete state spaces, it is postulated that the optimal policy can be represented in a format other than a table. The ACAS X program has funded various efforts to explore this idea. These efforts have focused on manipulating the state-space representation once the optimal policy has been generated, and have not addressed the underlying structure of the state-transitions themselves. These efforts do not address the algorithm design question and complement the described research.

3.2 Conceptualizing the Aircraft Avoidance Problem

An aircraft encounter occurs when two or more aircraft come within close proximity of each other and ATC separation has failed and can not provide separation assistance. The avoidance problem is then postulated as two simple, yet fundamental, questions:

1. What is the risk of a loss of separation and collision with another aircraft?
2. If the collision risk is sufficient, what is the optimal action to minimize risk?

While the first question solely addresses safety as defined by some loss of separation, such as an NMAC, the second question defines optimality via both safety and operational constraints. An algorithm that alerts at even the slightest risk of NMAC would be very safe but would not be considered optimal because it doesn't meet operational alert rate requirements. Conceptually then the aircraft avoidance problem is composed of safety and operational elements. The safety element primarily addressed the first question and the operational element the second. Based on this conceptualization, an MDP can be organized.

3.2.1 Safety

The safety problem quantifies the risk of collision and is easily defined numerically by three metrics:

- Horizontal and vertical distance at CPA; and
- Time to CPA.

A NMAC¹ is used as a surrogate for collisions. It is a binary state, a system of multiple aircraft are either in an NMAC or not. Time to CPA is not binary and quantifies the risk of transiting to an NMAC. Time to CPA is also used to classify collision avoidance encounters, defined as when time to CPA is less than 45 – 60 s. This is due to the amount of time required to increase separation to avoid an NMAC, given ownship's current state (Morrel, 1956).

¹Separation of ± 100 ft vertically and ± 500 ft horizontally

This time to CPA range is verified through simple calculations when only one aircraft is maneuvering to avoid an NMAC. First, consider the vertical axis:

$$\dot{h}_o \times t_v = 100 \text{ ft} \quad (3.1)$$

$$t_v \left(\dot{h}_o \right) = \frac{100 \text{ ft}}{\dot{h}_o} \quad (3.2)$$

$$t_v (25.0 \text{ ft/s}) = 4.0 \text{ s} \quad (3.3)$$

$$t_v (12.5 \text{ ft/s}) = 8.0 \text{ s} \quad (3.4)$$

Next, the required time to maneuver horizontally is slightly more complicated given bank angle (θ_h) and cross aircraft motion (A) (Wood, 1987).

$$\frac{A}{2} \left[32 \tan(\theta_h) \times t_{CPA_h}^2 \right] = 500 \text{ ft} \quad (3.5)$$

$$t_h(A, \theta_h) = \sqrt{\frac{1000A}{32 \tan(\theta_h)}} \quad (3.6)$$

$$t_h(1, 25^\circ) = 8.19 \text{ s} \quad (3.7)$$

$$t_h(1, 15^\circ) = 10.8 \text{ s} \quad (3.8)$$

A is generally less than one and t_h will be greater since cross aircraft motion is only parallel to the horizontal miss distance vector when the two aircraft are flying in exactly opposite directions. An algorithm should not just narrowly prevent an NMAC but increase separation as well. Extending the equations to account for this yields maneuver times between 30 – 60 s. Time to CPA can be viewed as a surrogate for the risk of a undesirable horizontal and vertical miss distance at CPA. A greater time to CPA represent less risk of eventually transitioning to an NMAC in the future than a smaller time to CPA.

3.2.2 Operational

Unlike safety, the operational problem is not clearly defined numerically or easily decomposed into mathematical equations. This presents a challenging problem in optimization because a specific state can not be designated as “bad” or “good” using a binary cost structure, like NMAC. Any NMAC can be avoided by increasing separation in either the vertical axis or horizontal plane given sufficient time prior to CPA, the operational problem decomposes into three questions:

- How much does each action reduce the safety risk?
- What is the smallest time to CPA to initiate each action?
- What is the expected time to complete each maneuver?

Generally the first question is answered by how much does each action increase or decrease separation. An action should only be selected when the probability of the encounter resolving itself without an RA is low. The third question introduces a subtle, yet complex, component. Short alerts may appear unneeded and annoying while longer alerts may appear tedious or significantly impair the aircraft’s mission.

The operational problem is mostly aircraft class or platform specific, whereas the safety problem is aircraft class agnostic. The NMAC definition is the same for all aircraft but operational considerations can vary widely between aircraft. Operational optimality is a function of an aircraft’s performance (i.e. maximum vertical rates), mission objectives, and operating environment. During the development of TCAS, this operational problem wasn’t as prevalent because it was assumed that only manned commercial aircraft would equip TCAS. The introduction of UAS into the airspace and desire to equip general aviation aircraft with collision avoidance, has increased this complexity and relevancy of the operational problem.

The operational problem is not independent on the safety problem and is dependent upon time to CPA. This is evident by the use of τ as a countdown to CPA and ACAS X defining its initial optimization value using τ .

3.3 NMAC Entropy

Collision avoidance is naturally defined by two events: in an NMAC or not an NMAC. By determining the probability of these events through simulation for each state transition, NMAC entropy can be calculated. This quantifies the measure of NMAC randomness for a given state. By summing over all state-transitions, the total NMAC entropy for a specific state is calculated. Since entropy is superadditive, NMAC entropy for each state can be calculated individually.

$$H = \left[P(\text{NMAC} = 1) \log_2 (P(\text{NMAC} = 1)) \right] + \left[P(\text{NMAC} = 0) \log_2 (P(\text{NMAC} = 0)) \right] \quad (3.9)$$

The memory requirements for each state is paired with it's NMAC entropy, allowing the calculation of entropy per memory. This enables an understanding the computational efficiency of the state. Comparing the entropy of two different states leads to an analytic and quantitative approach for selecting a state space based on NMAC. Conceptually, if a state has low NMAC entropy then will likely not provide sufficient information about the potential of transitioning to an NMAC. It is postulated that a state space with greater NMAC entropy will perform better than a state space with less NMAC entropy, given some assumptions. Foremost, the states can not all serve the same categorical function. For example, a state space consist of only vertical rates states and aggregate features does not provide sufficient information about encounter geometry regardless of the entropy sum. To facilitate this state exploration, for each state, various uniformed discretizations were simulated with an

all-inclusive range that was determined via simulation.

NMAC entropy will also be used to quantitatively decide the order in which states are added to the state space. Starting with a single state, the question “what state should be added next?” will be answered using information gain and the Kullback-Leibler divergence (Kullback, 1997). Generally, a state with high mutual information should be preferred. The state that provides the greatest gain, will be the one added to the state space. Since a state’s NMAC entropy is dependent upon the state’s discretization, building a state space using information theory enables the comparison of states across discretizations. Information gain and the Kullback-Leibler divergence provide a quantitative analysis between states. For example, it is possible that a course discretized state provides greater information regarding NMAC than another finely discretized state. This is an alternative to the time intensive approach of generating optimal solutions for various state spaces using different costs and evaluating against safety and operational metrics. This information theory approach will help identify potentially inefficient state spaces before MDP optimization and evaluation.

3.4 Action Space

Foremost, the potential actions than are available to any aircraft must be determined and then filtered to define an action set used to avoid an intruder. With the advent of UASs, it can no longer be assumed that aircraft are controlled in a similar manner. For just UASs, there are four different levels of horizontal aircraft control, four levels of vertical control, and three levels of speed control (Williams, 2007). The lowest level with direct control is a traditional pilot’s yoke while the highest level is programmed waypoints with no direct control. A simple example highlights the wide variety of potential actions. Assume an aircraft needs to climb to an altitude of 6500 ft, a small subset of potential controls are:

- Issue a waypoint of 6500 ft,
- Climb at X ft/min,
- Climb to an altitude of 6500 ft,
- Generate a pitch rate of Y deg/s, and
- Climb to an altitude of 6500 ft at a rate of X ft/min,
- Generate a pitch acceleration of Z ft/s².

Depending on the selected action, the aircraft can behave in different ways; vertical take-off and landing (VTOL), fixed-wing and airships all have different performance characteristics and constraints. Issuing a fixed vertical rate is more deterministic than issuing a desired altitude which can be achieving by a wide variety of vertical rates. For a traditional DP framework, different state dynamics would need to exist for the different controls. Under the simulation-based approach, the actions can be observed via the simulation output and recorded. Changing controls is a matter of selecting different states and regenerating the state-transition and cost matrices; code development for each action is not required.

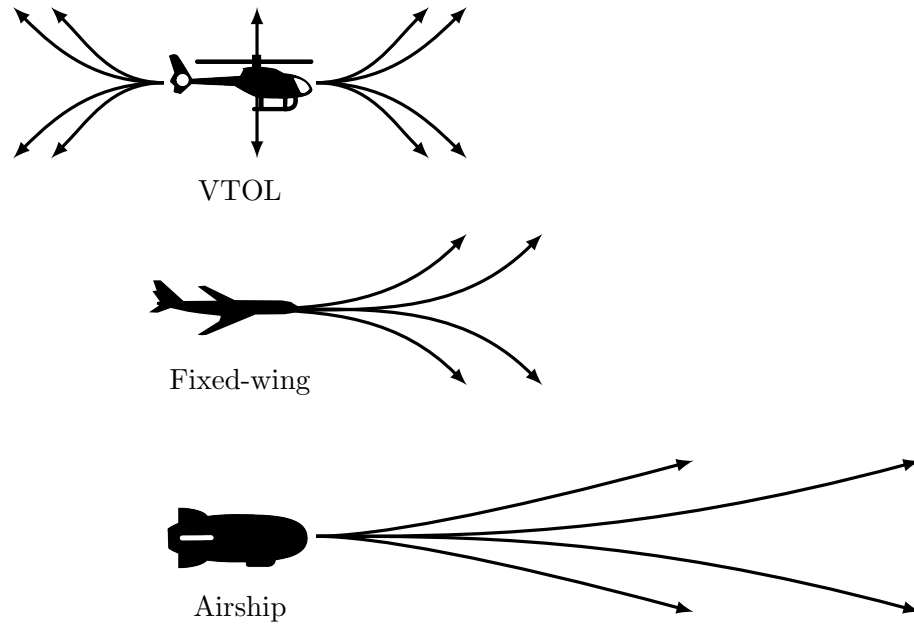


Figure 3.1: Example aircraft vertical action variety

The simulation-based framework supports any combination of potential controls; however, some actions have been historically validated to be effective. Avoidance maneuvers via a change in airspeed is not one of these and were not considered. Airspeed has the least effect on encounter geometry. Over the course of 30 – 60 s airspeed can only nominally be changed. For the simplistic case of a head-on encounter, a change in airspeed will have a minimal effect on TCA and on the encounter geometry. Changing airspeed in hopes of having the intruder either pass in front or below of the ownship isn't effective.

Unlike changes in airspeed, vertical maneuvers via commanded vertical rates have been shown to be historically effective, as demonstrated by TCAS RAs. Thus a vertical action set representing manned aircraft consists of single climb and descend actions of $\pm 25 \text{ ft/s}$ ($\pm 1500 \text{ ft/min}$). These rates adequately increase separation without giving the unknowing passengers the distinct feeling of a severe maneuver. A TCAS designer described this decision as, “we didn't want to spill the martinis.” However as noted in Section 2.3.1, assumptions for these rates are not appropriate for emerging user classes, such as UASs. A second set of vertical rates of $\pm 12.5 \text{ ft/s}$ ($\pm 750 \text{ ft/min}$) represents UASs and other limited performance aircraft.

While vertical aircraft avoidance has been researched for over fifty years, horizontal maneuvers have only recently been considered a feasible maneuver. This is contributed to better surveillance technology. The horizontal action space is based on a standard rate one turn (ROT) and is a function of true airspeed and bank angle, as shown in Figure 3-2. . For light aircraft, it is defined as a 3 deg/s turn, which completes a 360 deg turn in 2 min. It is assumed that all aircraft can achieve a standard ROT and is not perceived as abnormal behavior to ATC.

For relatively fast airspeeds, a bank angle greater than 30 deg is required for a standard ROT. This may not be operational feasible. For example, a manned Boeing

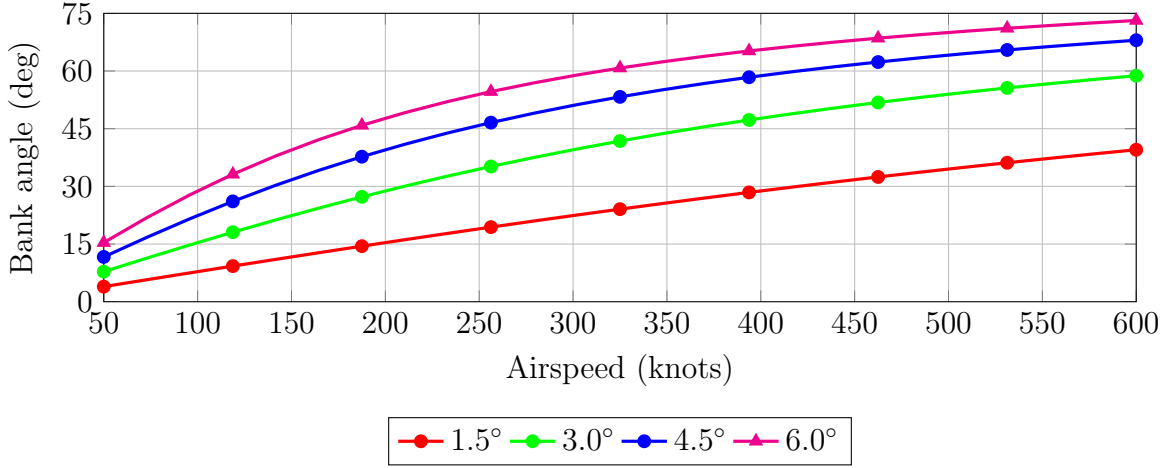


Figure 3-2: Required airspeed and bank angle for various turn rates.

777 has a cruise speed of approximately 825 ft/sec (490 kt), which requires greater than a 45 deg bank angle to achieve at standard ROT. Additionally, high bank angles at slow airspeeds can result in the loss of control for some aircraft. Hence, rate half turns (1.5 deg/s) are normally used when airspeed is greater than 420 ft/s (250 kt).

Tables 3.1 – 3.2 summarize the action sets and assume that both aircraft are fixed-wing. Rotorcraft and VTOLs which can hover or airships with extremely slow airspeeds are not considered. There is no distinction made if one action has a stronger sense than another. An alerted action only affects the corresponding state. For example, if an aircraft is nominally climbing and a horizontal alert is issued, the aircraft will execute the alert maneuver while climbing. To reduce problem complexity, strengthening will not be considered and reversals will be limited. Strengthening alerts are more disruptive with greater rates and are designed if the initial RA is not increasing separation fast enough. The simulation-based framework can support additional actions, if they are observed within the Monte Carlo simulation.

Actions that control both horizontal and vertical states were not considered. An example is “straight and level” which sets vertical rate to 0 ft/min and turn rate to 0 deg/s. This varies from a preventive RA because it is actively controlling the rates.

Representative action sets.

Table 3.1: Manned.

Command	Description
COC	Clear of conflict
CL1500	Climb at 1500 ft/min
DE1500	Descend at 1500 ft/min
L3	Light turn at 3 deg/s
R3	Right turn at 3 deg/s

Table 3.2: UAS.

Command	Description
COC	Clear of conflict
CL750	Climb at 750 ft/min
DE750	Descend at 750 ft/min
L3	Light turn at 3 deg/s
R3	Right turn at 3 deg/s

3.5 State Space

The state space represents the state of the world. Prior to selecting states an understanding of coordinate systems need to be established to spacial frame of reference. Next the relationship between states and NMAC are identified. Finally, the relationship between states are explored to identify potential memory savings and state flexibility.

Aircraft are commonly represented in three-dimensional Euclidean space. Figure 3-3 illustrate the three common coordinate systems used to represent points in three-dimensional Euclidean space: Cartesian, cylindrical, and spherical. The Cartesian system specifies a point uniquely in a plane by a triple of numerical coordinates. The cylindrical system specifies a point by the distance from a chosen reference axis, the direction from the axis relative to a chosen reference direction, and the distance from a chosen reference plane perpendicular to the axis. The spherical system specifies a point by the radial distance from a fixed origin along with horizontal and vertical angles. An immediate disadvantage of the spherical system is that altitude can't be represented as a single variable.

Physically, aircraft are observed either by primary or secondary radar. Primary radar transmits a pulse of radio energy and generates an aircraft position estimate based on the reflected energy. It operates independently of the target aircraft. Pri-

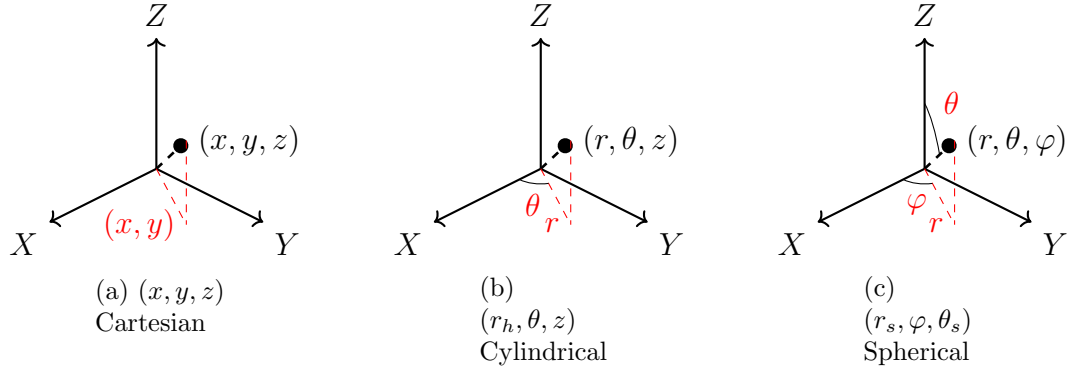


Figure 3.3: Real number coordinate systems

many radar often uses either a spherical or cylindrical coordinate system. Secondary radar, such as Mode S, relies on equipped radar transponder, which transmits a response to each intruder's interrogation signal. Secondary radar traditionally use the cylindrical coordinate system but recent technology, such as ADS-B, leverage a Cartesian coordinate system. In application, certain states are deemed to provide more utility such as discretized Cartesian altitude provided by beacon surveillance.

The NMAC definition nicely maps to a relative cylindrical coordinate system with the horizontal range angle ignored by the NMAC definition. How the encounter transitions to an NMAC is quantified by relative range rate $\Delta \dot{r}$ and relative vertical rate $\Delta \dot{h}$. These rates are dependent upon all aircraft in the encounter and a single aircraft can't directly control these relative rates.

As discussed in Section 3.4, there many different actions an aircraft can implement. Different actions will affect these rates differently and subsequently the potential to transition to an NMAC-related state. There is a difference between commanded actions and how the aircraft responds. Therefore when selecting a state space, an attempt to avoid leveraging a state that directly ties to only a subset of potential actions is made. For example, if the horizontal action is commanded turn rate, the turn radius will be dependent upon airspeed. The turn radius will influence the

potential transition to an NMAC. Yet, current turn rate isn't ideal because it doesn't quantify an aircraft that uses waypoints.

The state set must contain sufficient information to represent how each action influence the state-transitions probabilities. Environmental variables such as wind or atmospheric density are not considered to be sufficient information. It is assumed that aircraft are close enough to be experiencing the same environmental condition. For example, a climb or descend will change the \dot{h} and Δh states and associated transition probabilities. An 11-dimensional tuple state space has been previously proposed but was recognized as impractical due to dimensionality (Kochenderfer et al., 2010a):

- relative altitude Δh ,
- horizontal range r_h ,
- own vertical rate \dot{h}_o ,
- intruder vertical rate \dot{h}_i ,
- own ground speed v_o ,
- intruder ground speed v_i ,
- bearing of intruder α ,
- relative heading $\Delta\psi$,
- own turn rate ϕ_o ,
- intruder turn rate ϕ_i , and
- advisory state s_{RA} .

A grid representation of this state space with 21 edges per dimension would result in 3.5×10^{14} discrete states and is infeasible for DP optimization. Instead of decomposing into τ states to reduce dimensionality, the three-dimensional coordinate system can be viewed as a vertical axis and a horizontal plane and treated independently. Each independent space can then be explored for both traditional first-order states and new aggregate features.

3.5.1 System Perspective

Aircraft algorithm design is typically viewed from ownship’s perspective and presents the intruder as an uncontrolled aircraft some distance away. During coordinated maneuvers, an action is recommended to the intruder aircraft, but there is no guarantee that the either aircraft will comply. This can be either due to latency, negligence, or physical constraints on the aircraft. Section 2.3.3 and the ACAS Xu design highlight this problem. A sufficient state space requires both understanding of the controlled aircraft, the intruder aircraft, and the relative position of the intruder from the perspective of ownship.

An NMAC event requires multiple aircraft, no aircraft can be in an NMAC alone. Using an NMAC-centric perspective then could reduce the state space requirements. This perspective is very similar to the traditional relative to the ownhsip but emphasize the relationship between the aircraft. The success of τ shows the benefit of modeling collision avoidance from an system perspective, rather than individual aircraft. Range and range rate are functions of both aircraft and it directly correlates to NMAC risk. Emphasizing relationship naturally leads to the development of aggregate features.

There are many different types of space aggregation, the implemented aggregation constructs a nonlinear, piecewise constant feature-based architecture. For native state is partitioned based on “similar” features between states. When extracting features from this space, the original state is associated with unique aggregate states and subsets. Discretized MDP states are then generated from these features. States are often generated through simple operations (i.e. summation) across features. Alternative space aggregation approaches will associate the aggregate states with the original space or both.

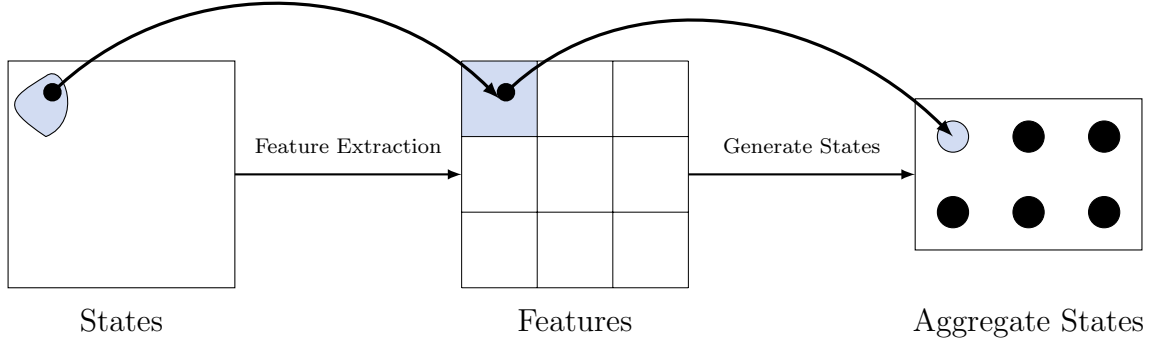


Figure 3.4: Feature-based space aggregation (Bertsekas, 2005).

3.5.2 Vertical Axis

Vertical separation (Δh) directly corresponds to the NMAC definition and notion of safety. The vertical rates directly relate to potential actions that can increase vertical separation and reduce the safety risk. Since the vertical axis is a line, calculating the vertical rate change ($\dot{\Delta h}$) is easy. It is defined by vertical separation (Δh) and the aircraft's vertical rates; it is an aggregate feature that quantifies $(\dot{h}_o, \dot{h}_i \mid \Delta h)$:

$$\dot{\Delta h}(\dot{h}_o, \dot{h}_i \mid \Delta h) = \begin{cases} \dot{h}_o + |\dot{h}_i| & \Delta h \geq 0, \dot{h}_i \leq 0 \\ \dot{h}_o - |\dot{h}_i| & \Delta h \geq 0, \dot{h}_i > 0 \\ \dot{h}_i + |\dot{h}_o| & \Delta h < 0, \dot{h}_o \leq 0 \\ \dot{h}_i - |\dot{h}_o| & \Delta h < 0, \dot{h}_o > 0 \end{cases} \quad (3.10)$$

Figure 3.5 illustrates these three states, which provide sufficient information to represent where the aircraft are vertically spaced (Δh) and the rate at which their locations are changing (\dot{h}_0, \dot{h}_i). Both TCAS and ACAS Xa leverage them. However, there are other potential states that may provide greater utility. Historical success of these states does not preclude potential success of other states.

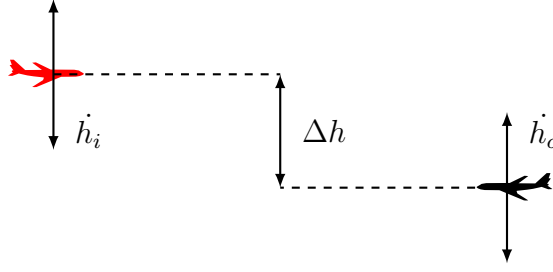


Figure 3-5: First-order vertical states.

While useful, \dot{h}_i alone doesn't directly explain how the overall encounter system is evolving. Consider the pair $(\dot{h}_o = -25, \dot{h}_i = 0)$ where the ownship is the only aircraft affecting Δh . Depending on the encounter geometry, a level-off preventive RA or reversal to a corrective climb RA maybe optimal. Rather how the different aircraft vertical rates interplay is more important. The $\Delta \dot{h}$ state quantifies this interaction but is also more complicated, it is hypothesized that simpler aggregate feature can approximate $\Delta \dot{h}$ and better quantify NMAC risk than the first-order state \dot{h}_i .

Aggregating Rates

A simple aggregate feature, such as $(\dot{h}_o + \dot{h}_i)$, can represent different states given \dot{h}_o with a single value. For example, as illustrated by Figure 3-6, $(\dot{h}_o + \dot{h}_i) | \dot{h}_o = 0$, quantifies if both aircraft are in level flight but also for when aircraft have opposite vertical senses.

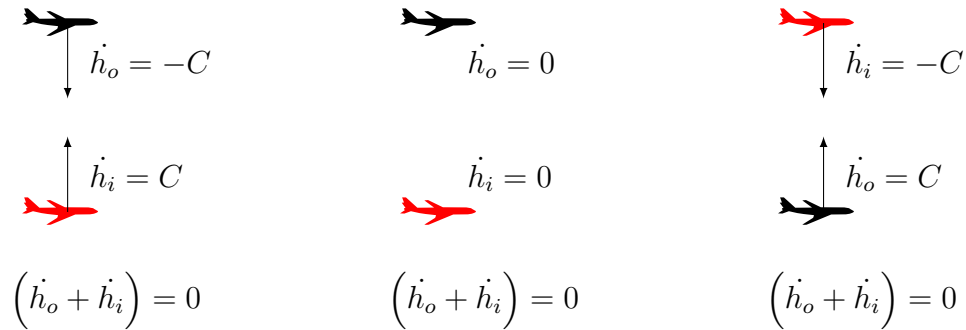


Figure 3-6: Simple visualization of vertical rate states

If a state space included this state and \dot{h}_o , then its possible to distinguish between the different vertical rate combinations. While $\dot{h}_o = \{-C, 0, C\}$, \dot{h}_i requires three values each for a total of nine elements to describe Figure 3-6, $(\dot{h}_o + \dot{h}_i)$ requires only a single state for a total of three elements in the space. Although the reduction of two variables seems minuet, it can lead to exponential memory savings due to the combinational nature of DP.

The interaction between \dot{h}_o , \dot{h}_i , $(\dot{h}_o + \dot{h}_i)$ can be quantified via simple simulation where the rates range from -50 ft/s to 50 ft/s with a uniform discretization of 1 ft/s. There are 10,201 different combinations but only 201 unique $(\dot{h}_o + \dot{h}_i)$ values and 75% of combinations produce a $(\dot{h}_o + \dot{h}_i)$ within the vertical rates range of ± 50 ft/sec, shown in Figure 3-7. More importantly, $(\dot{h}_o + \dot{h}_i)$ is not explicitly bounded by \dot{h}_o .

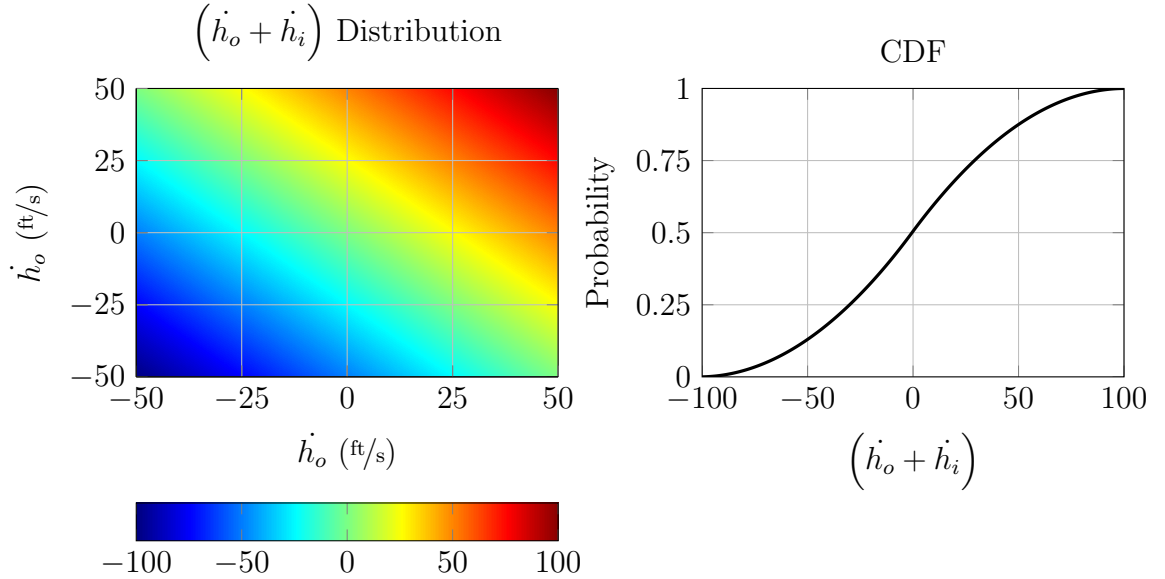


Figure 3-7: $(\dot{h}_o + \dot{h}_i \mid \dot{h}_o, \dot{h}_i)$ distribution and CDF

ACAS Xa currently ranges \dot{h}_o and \dot{h}_i from ± 42 ft/s, however there are aircraft that can exceed those vertical rates, such as military aircraft. Depending on the bound of

$(\dot{h}_o + \dot{h}_i)$, it can quantify a \dot{h}_i outside the range of \dot{h}_o .

$$\begin{aligned} (\dot{h}_o + \dot{h}_i \mid \dot{h}_o = -25) &\in \{-42, \dots, 42\} && \iff -17 \leq \dot{h}_i \leq 67 \\ (\dot{h}_o + \dot{h}_i \mid \dot{h}_o = -25) &\in \{-84, \dots, 84\} && \iff -59 \leq \dot{h}_i \leq 109 \end{aligned}$$

Since majority of $(\dot{h}_o + \dot{h}_i)$ values occur within the bounds of the vertical rate, selecting a discretization that is memory efficient with an appropriate number of states is feasible. Furthermore, the probability of a high vertical rate is relatively low, thus there is no necessity to finely discretize outside the bounds of \dot{h}_o . This leads a reduction in memory, latter shown in Section 5.2.6. In addition to $(\dot{h}_o + \dot{h}_i)$, other aggregate features such as $(\dot{h}_o - \dot{h}_i)$ and various ratios of vertical rates were assessed.

Spherical Coordinates

A different option to the traditional cylindrical coordinate system representation of the vertical axis is the spherical system which measures vertical separation as an inclination angle (θ_s) given spherical range (r_s). Since r_h is significantly greater than Δh , r_s should be very similar to r_h . If r_s is large, then a vertical maneuver shouldn't be required in most cases and Δh isn't as important. Figure 3-8 illustrates this by calculating $\Delta h(r_s, \theta_s)$; as r_s increases, the range of Δh also increases.

It was hypothesized that since Δh is very large for modest θ_s values at even moderate r_s , that the full range of $\theta_s = \{0, \dots, 180\}$ isn't required. As r_s decreases, the range of potential transitions of θ_s would increase, leading to a dynamic discretization of vertical separation. Additionally θ_s is a first order output of many radar systems which potentially would help make the optimal policy more human readable.

However after review, it was deemed that using θ_s as part of the state space is not acceptable. Foremost, it also doesn't take advantage of beacon technology and the

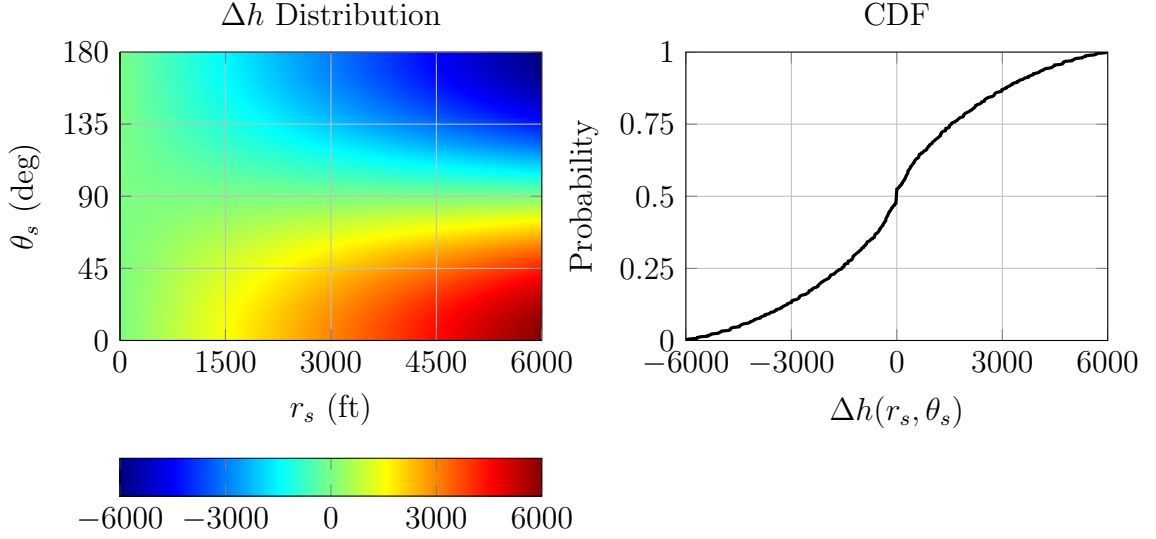


Figure 3-8: $\Delta h(r_s, \theta_s)$ distribution and CDF

quantized altitude measurements it provides, leading to unnecessary greater vertical uncertainty. Although the Δh between angles does dynamically change, the aircraft must be dangerously close to take advantage of the increased vertical fidelity. For r_s beyond a few miles, nearby θ_s states could correspond to hundreds of feet difference in Δh . This is emphasized when comparing angular states in the results Section 5.2.4.

3.5.3 Horizontal Plane

There are many more potential states to represent the horizontal plane. This is a consequence of horizontal being a plane and the vertical axis can be viewed as a line. First order states include an aircraft's airspeed and direction and the relative range and direction between aircraft. The τ_h state is an approximation of the separation vector between aircraft due to the poor directional (azimuthal) measurements. From an optimization perspective, each of these states have advantages and disadvantages.

Specifically, airspeed (v) is the magnitude of the an aircraft's vector in the horizontal plane. This applies to any simple transformations of these magnitudes, such

as $(v_o + v_i)$. The aircrafts' airspeeds are easy to observe but independently do not provide sufficient information if separation is increasing or decreasing. Range rate (\dot{r}_h) provides this information but doesn't distinguish how aircraft influence \dot{r}_h .

Magnitude Information

To illustrate this, consider a simple level altitude, head on encounter between two aircraft where the airspeed ranges from 0 ft/s to 1000 ft/s (approximately 600 kts) with a uniform discretization of 20 ft/s. Figure 3.9 illustrates the range rate produced by any combination of airspeeds. Since this is a head-encounter with no acceleration, $\dot{r}_h = (v_o + v_i)$. Of the 2500 airspeed combinations, there are only 162 unique \dot{r}_h states. The small percentage of unique combinations indicate that a space including $\{v_o, v_i\}$ may not be memory efficient. This is important for considering how different combinations of (v_o, v_i) influence horizontal separation and subsequently NMAC risk.

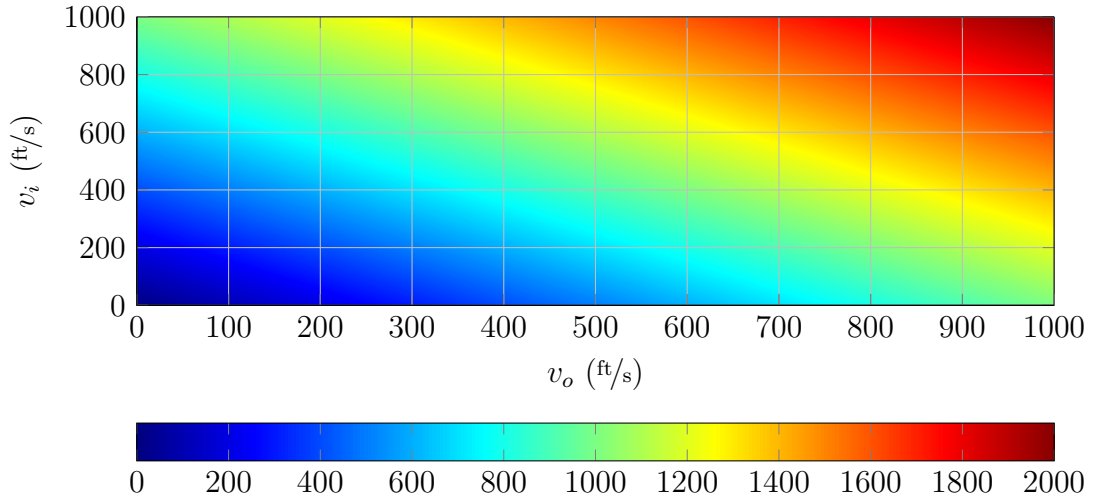


Figure 3.9: $\dot{r}_h(v_o, v_i)$ for head-on encounter

Figure 3.10 illustrates the corresponding τ_h distribution and CDF given a head-on encounter at $r_h = 30381$ (5 nm). A max τ_h of 195 s is assumed; this maximum is larger than what TCAS or ACAS Xa assume. 75% of airspeed combinations have a τ_h less than 45 s which is approximately a quarter of the maximum. Even if r_h is extended to

10 nm, 75% of τ_h values would be less than 85 s, which is still less than half of the 195 s maximum. The long tail of the $\tau_h(v_o, v_i)$ distribution presents a key optimization challenge and the maximum τ_h dictates how saturated the τ_h distribution will be.

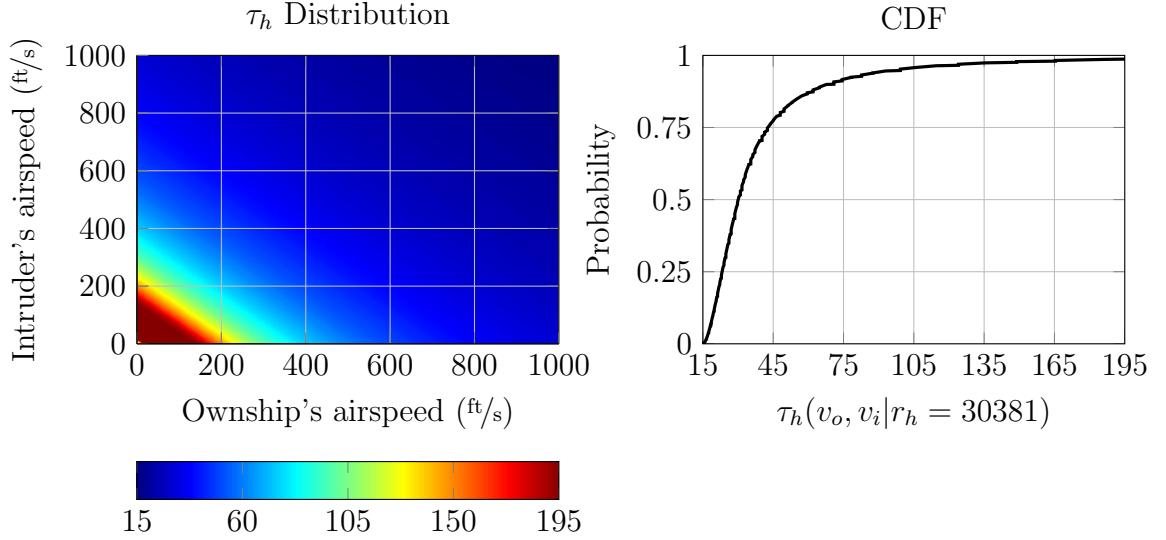


Figure 3-10: $\tau_h(v_o, v_i | r_h = 30381)$ distribution and CDF

The τ_h tail is a function of the large range of (v_o, v_i) and contrasts the relatively small range of (h_o, h_i) . Understanding the relationship between specific airspeed states and NMAC are critical for developing an airspeed discretization. Specifically what is the NMAC risk for each state and if is the relative risk great enough between states to warrant inclusion.

Angular Information

Directional information is required to form a complete vector. These vectors can be relative or absolute. Relative heading ($\Delta\psi$), the horizontal resultant vector angle ($\psi_{\vec{R}}$) and bearing (α) are the three angular states considered.

$$\Delta\psi = \psi_o - \psi_i \quad (3.11)$$

$$\psi_{\bar{R}} = \tan^{-1} \left[\frac{(v_o \times \cos(\psi_o)) + (v_i \times \cos(\psi_i))}{(v_o \times \sin(\psi_o)) + (v_i \times \sin(\psi_i))} \right] \quad (3.12)$$

$$\alpha = \tan 2^{-1} [(y_o - y_i), (x_o - x_i)] \quad (3.13)$$

Bearing (α) is calculated using a $\tan 2^{-1}(y, x)$ which calculates with respect to the complex number $x + iy$. In practice it is used by a variety of computer languages to calculate the arctangent with respect the appropriate quadrant of the computed angle (Organick, 1966). By taking into account the quadrant, bearing (α) is able to distinguish between left/right and front/behind relative geometries.

For a given r_h , the relative geometry defines the encounter type, which impacts safety risk, regulatory rules, and optimal maneuvers. Figure 3-11 illustrates that although the encounter geometry is different with each intruder aircraft, only bearing (α) varies. Bearing should provide more information about how a system of two aircraft evolve into an NMAC with the same or less memory requirements. However, bearing doesn't take into account individual aircraft vector angles, a head-on encounter and an ownship directly overtaking an intruder encounter will produce the same value, whereas $\Delta\psi$ and $\psi_{\bar{R}}$ will distinguish between these two encounter types.

Vector Representation

Consider an encounter sampled from the correlated encounter model in Figure 3-12. The ownship is traveling straight and level while the intruder is circling in a holding pattern and neither aircraft accelerate. The initial r_h is 13150 ft (approximately 2.1 nm) and the intruder will turn into the ownship from the ownship's left. TCA occurs at 38 s with an NMAC due to 47 ft and 494 ft of vertical and horizontal separation.

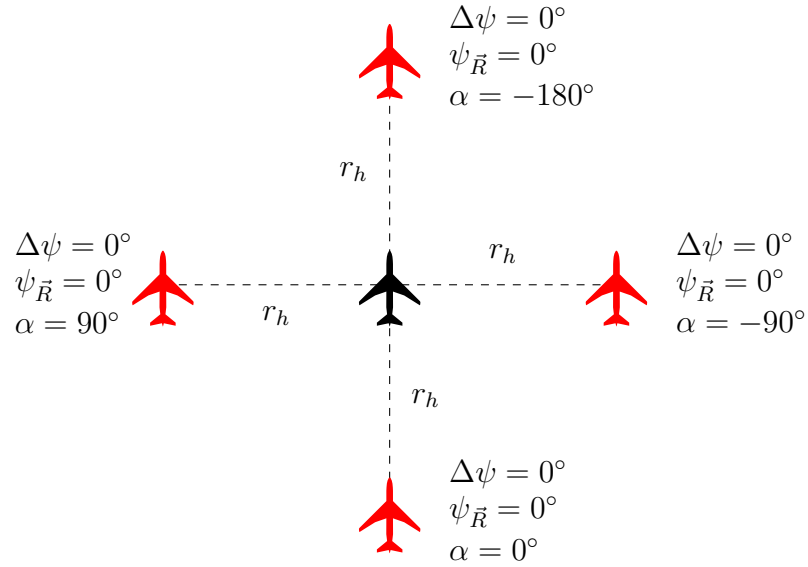


Figure 3-11: Simple visualization of angular states

Figure 3-13 illustrates r_h , \dot{r}_h , various airspeed states, and various angular states of the encounter in Figure 3-12. Foremost the three angular states of relative heading ($\Delta\psi$), bearing, and resultant vector angle ($\psi_{\vec{R}}$) evolve differently. Only bearing (α) experiences a sharp change near TCA and changes its behavior based on r_h .

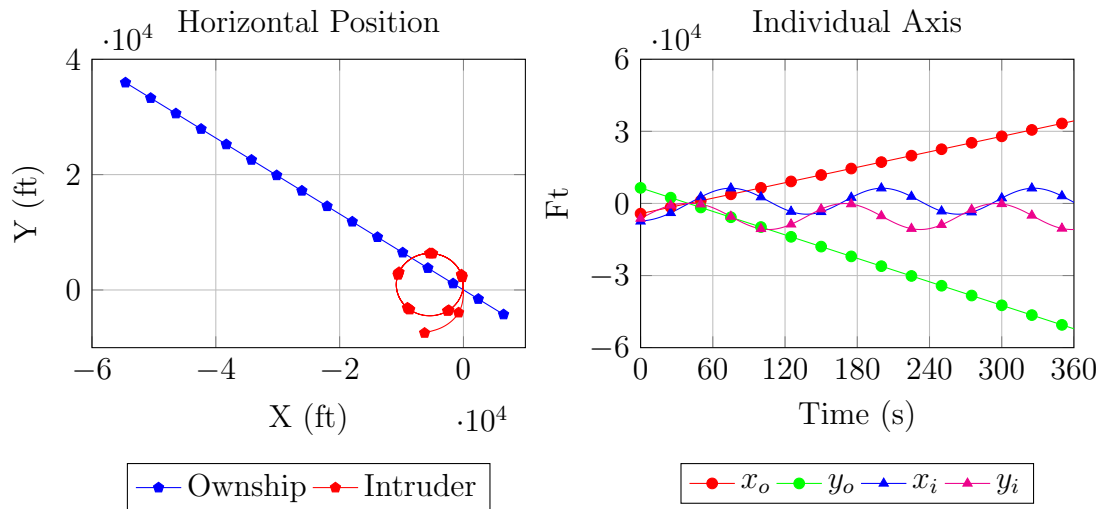


Figure 3-12: Example encounter

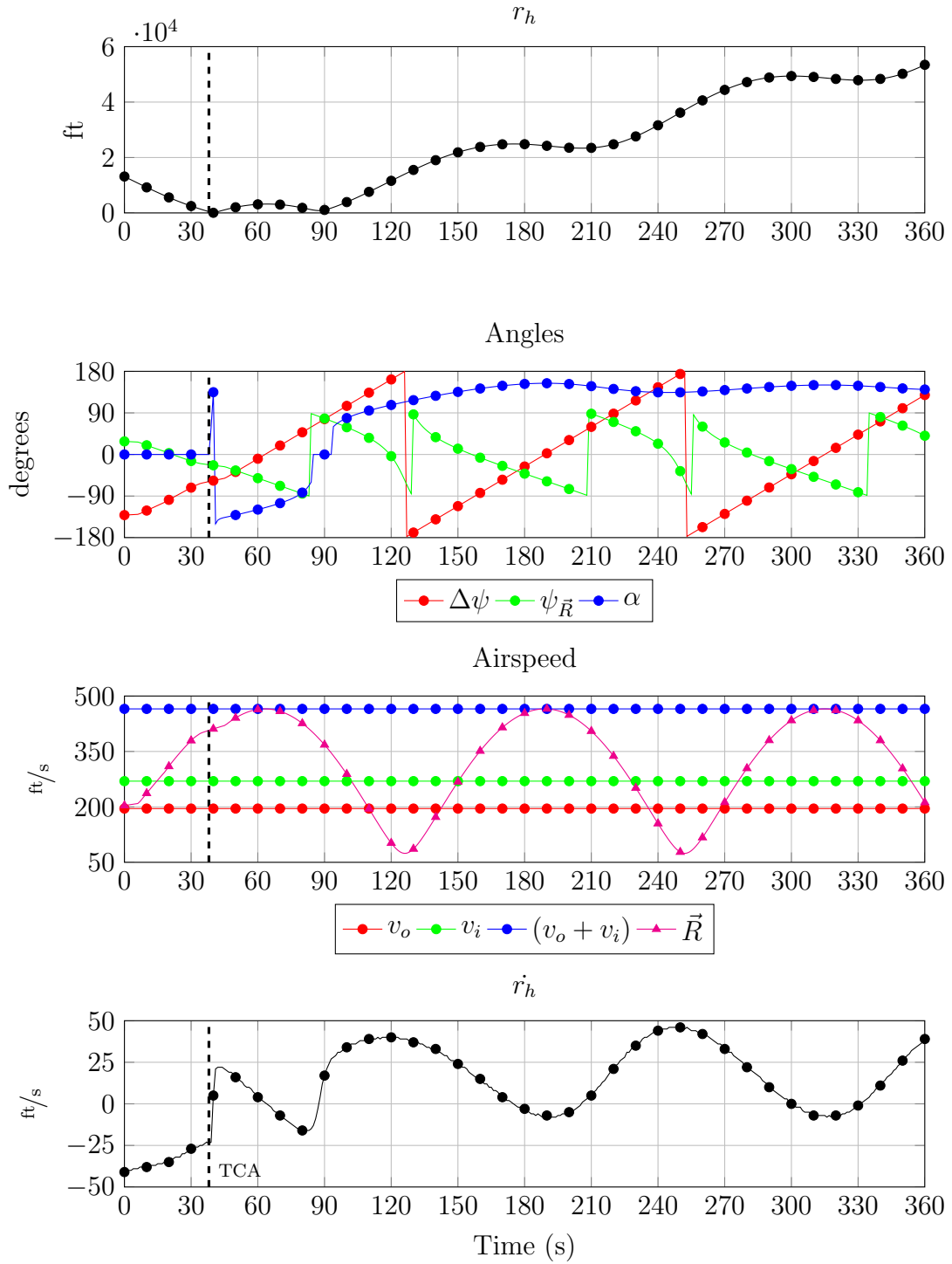


Figure 3-13: Various states for example encounter in Figure 3-12

The traditional first-order airspeed states of ownship (v_o) and intruder (v_i) airspeeds are static throughout the encounter since their no airspeed acceleration. The aggregate feature of these states ($v_o + v_i$) must also be static then. Only the resultant vector magnitude (\vec{R}) and range rate (r_h) change as the encounter evolves. Since \vec{R} is calculated using angular information, it correlates with $\Delta\psi$ and $\psi_{\vec{R}}$. Only r_h indicates if separation is increasing or decreasing while having some correlation with the angular rates.

3.6 MDP Cost

As described in Section 2.1, each element in the cost matrix is associated with the corresponding element in the state-transition matrix. The optimal policy’s behavior is directly influenced by these costs, however the interaction between various costs are not always obvious. ACAS X has experimented with surrogate models (Smith, 2013) and online costs (Asmar and Kochenderfer, 2013) to better manipulate and understand MDP costs. Neither of these are leveraged because they do not directly support the research objectives. Using a cost equation dependent upon the state-transitions and actions have been shown to be sufficient.

3.6.1 NMAC Horizon

Since ACAS X models transition as a white-noise process, it doesn’t directly consider the frequency of each state-transition. Instead it focuses on calculating the NMAC risk at each state and avoiding a very small “bad” region, NMAC. Then choosing whether to select an undesirable alert / action that reduces the risk of transitioning into NMAC. The safety problem is quantified through an NMAC binary cost and a few other costs to promote increasing separation near an NMAC. Majority of the costs then quantify the operational problem and influence if alerting is more preferable than increasing the NMAC risk.

This approach calculates an alerting region over τ where the maximum τ value is always clear of conflict (COC). The DP algorithm determines for a given τ and other state variables if an alert is required. τ is not robust to specific types of encounters, such as parallel encounters where two aircraft are flying in parallel. Regardless of r_h , a parallel formation results in $\dot{r}_h = 0$, which to a maximum τ . To address this, ACAS X implements a procedure mode optimized via surrogate modeling (Smith, 2013).

An alternative approach is to define a “good” region and provide costs to select actions that transition to it, instead of attempting to transition away from a “bad” region defined by an NMAC. Although the end goal of increasing separation is the same, how the costs are implemented are different. Since an aircraft encounter is defined by time to CPA and can be used a surrogate for NMAC risk, this alternative formulation uses cost-shaping to encourage alerting in a specific region, as defined by state-transition’s risk of transitioning to an NMAC within some time horizon. Through simulation, the one-step probability of transiting to an NMAC is calculated and the n -step probability of transiting to an NMAC. The simulation can estimate the probability of incurring an NMAC cost at some time horizon in the future. This binary state is named ν_t where t is the time horizon into the future and is true if an NMAC occurs within t time steps from current state and zero if not.

Defining alerting regions via simulation, is an extension of using simulations to establish a risk-based separation standard for UAS (Weibel et al., 2011). The previous research only calculated the probability of NMAC given τ or NMAC and a state. The newly developed simulation framework is not limited to τ and calculates any state given any other state, such as $P(\text{NMAC} \mid \Delta h, r_h)$. Furthermore, ν_t provides the DP algorithm additional information on which maneuvers increase separation the most over multiple time steps, instead of the single time step of the algorithm. This is

important when comparing vertical and horizontal maneuvers.

It is a critical feature of the described approach because it enables costs to be applied to states that don't include either Δh or r_h , the two states that directly define NMAC. This is incredibly important in evaluating a state-space using the spherical coordinate system or includes aggregate features composed of either Δh or r_h . The probability of NMAC for the state-transition pair (s, s') is denoted as $\lambda(s, s')$. Given a set of actions, the optimal control should minimize λ and reduce the potential of incurring the NMAC cost at a future state.

It is postulated that the conditional probability for each value of ν_t can be used a surrogate for the τ used by TCAS and ACAS X. Greater probabilities of $\nu_t = 1$ indicate that the current state is at higher risk of incurring the NMAC cost and corresponds to a lower traditional τ value. Then an alerting region can be defined using ν_t as a form of cost shaping. If there is little risk of an NMAC within some time horizon, then the algorithm doesn't need to alert.

3.6.2 Cost Functions

The cost functions are defined as indicator (characteristic) functions which can only take on a value of one or zero. They denote membership of an element of a subset A in the set X , where a value of one indicates membership (Cormen et al., 2001).

$$\mathbb{1}_A : X \rightarrow \{0, 1\} \tag{3.14}$$

NMAC cost

The primary cost function penalizes a potential NMAC state. As the avoidance policy will be evaluated based on how well it minimizes the risk of NMACs, the magnitude of the cost is relatively high compared to others. The NMAC cost $g(s, s')_{NMAC}$ as an

indicator function multiplied with some configurable time-invariant constant C_N .

$$g(s, s')_{NMAC} = \begin{cases} \mathbb{1}\{|\Delta h_{s,s'}| \leq 100, |r_{h,s,s'}| \leq 500\} \times C_N & \text{if } \Delta h, r_h \in S \\ \mathbb{1}\{\lambda(s, s') > 0\} \times C_N & \text{otherwise} \end{cases} \quad (3.15)$$

$$0 < |C_N| \quad (3.16)$$

Alert cost

A system that is extremely sensitive to other aircraft and alert unnecessarily is not operationally suitable. Collision avoidance alerts represent high-stress, time-critical interruptions that may distract users or ATC; unnecessary maneuvers over time leads to distrust of the system (Kuchar and Drumm, 2007). Operationally, COC is preferred because it is not interfering with current flight activities. Depending on operational considerations and platform capabilities, a different cost can be occurred for vertical or horizontal actions. They must be less than the NMAC penalty or the optimal control would never alert and actively transition out of the NMAC state.

$$\rho_v = \mathbb{1}\{a_k \neq \{\text{COC}\} | a_k = \{\text{CL25, DES25}\}\} \times C_{\alpha_v} \times \lambda_{ij} \quad (3.17)$$

$$\rho_h = \mathbb{1}\{a_k \neq \{\text{COC}\} | a_k = \{\text{L3, R3}\}\} \times C_{\alpha_h} \times \lambda_{ij} \quad (3.18)$$

$$0 < |C_{\rho_v}| < |C_\nu| \quad (3.19)$$

$$0 < |C_{\rho_h}| < |C_\nu| \quad (3.20)$$

ACAS X has an additional cost to discourage always issue a stronger RA. Since the action set, described in Section 3.4, does not include any stronger actions, a strengthening cost is not required.

COC reward

To help balance out the alert cost a small reward, the clear-of-conflict reward, is awarded at every time step the system is not alerting to provide some incentive to discontinue alerting after the encounter has been resolved. Similar to the others, this reward γ_k is an indicator function multiplied with some constant C_{COC} .

$$g(s, s')_{COC} = \mathbb{1}\{a = \text{COC}\} \times C_{COC} \quad (3.21)$$

Other potential costs

There are many other potential costs but to reduce problem complexity and to focus efforts on developing the simulation-based approach, other costs were not included. However, any cost can be implemented if the required information exists as part of the simulation. For discussion, two costs are presented here.

The reversal / strengthening cost is applied when not in COC and a different action is selected. This penalty discourages sharp transitions between action states. For example, it is disruptive to be in a climb advisory state and then commanded to descend. This cost also discourages yo-yo alerts, where the alert state fluctuates between multiple alerting states over a close set of timesteps (Holland et al., 2013).

Collision avoidance maneuvers are generally assumed to occur approximately one minute or less prior to CPA. With the integration UAS into the NAS, self-separation maneuvers are a focus of research. These maneuvers are attended to be less disruptive and to maintain well-clear whose criteria is still being formally defined.

3.7 Policy Evaluation

Implementing a MDP formulation and producing an optimal policy given cost functions are not sufficient to meet the research objectives. The optimal policy must produce reasonable decisions to mitigate aircraft collision risk. Monte Carlo simulations

have been historically used to evaluate the improvement in safety when leveraging a SAA system (Kuchar, 2005; Espindle et al., 2009; Kochenderfer et al., 2010c; Griffith and Edwards, 2010; Griffith and Olson, 2011; Edwards, 2012). Monte Carlo simulations provide exponentially more encounters to test against than equipping the policy onto physical hardware while being fiscally cheaper.

An encounter model simulation can be used to estimate the probability of a NMAC, which is a common safety metric used to evaluate aviation safety systems because it naturally relates to relevant safety criteria (e.g., collisions per flight hour). The equation for estimating the NMAC rate per flight hour λ_{NMAC} without faults is

$$\underbrace{P(\text{NMAC} \mid \text{encounter})}_{\substack{\text{estimate using} \\ \text{encounter model simulation}}} \times \lambda_{\text{encounters per flight hour}} . \quad (3.22)$$

The first term can be estimated using a simulation, such as that described in the previous section. The second term, $\lambda_{\text{encounters per flight hour}}$, can be obtained by observing air traffic to estimate how frequently aircraft encounter each other in the airspace.

A metric used to assess the relative benefit of various system configurations is the risk ratio. A risk ratio is the relative probability of a NMAC occurring given two different configurations and is a useful metric for comparing the relative performance of a collision avoidance system to a different type of system or to no collision avoidance system. For example, the relative benefit of equipping an avoidance capability compared to no capability is:

$$\frac{P(\text{NMAC} \mid \text{with the system})}{P(\text{NMAC} \mid \text{without the system})} . \quad (3.23)$$

If the risk ratio is less than one, then the system reduces the probability of a NMAC; if greater than one, the system increases the probability of a NMAC. A policy is generally determined to be feasible if the risk ratio is 0.1 or less.

Although the risk ratio is the primary safety metric, it is important to note that other metrics are valuable in operational assessments of an SAA system. Such metrics may include right of way compliance, alert rate, deviation from course, alert magnitude, etc. These metrics generally attempt to assess the operational suitability of the SAA system. The balance between safety and operational suitability will drive selection of the appropriate operating point for the SAA logic. While these metrics are outside the scope, they are critical to understanding overall SAA performance and can be assessed using the same analysis framework. Many of these require specific logic to evaluate so they are not presented here. Also note that the SAA workshop has recommended a target level of safety threshold, but one has not been sanctioned yet (Federal Aviation Administration, 2013).

Chapter 4

Implementation

A six step process was developed to implement the simulation-based framework and solve the MDP formulation discussed in Chapter 3. The Lincoln Laboratory Grid (LLGrid), a parallel computing system, and the pMatlab library (Bliss and Kepner, 2007; Kepner, 2009) were leveraged to supplement computational power.

1. Conduct Monte Carlo aircraft simulations and record all simulation states
2. Process raw simulation data into triples for D4M use
3. Calculate one-step state-transition probabilities assisted by D4M
4. Generate one-step cost matrix based on state-transitions
5. Optimize using DP
6. Evaluate optimal policy via Monte Carlo simulations

Aircraft encounters were simulated using the Collision Avoidance System Safety Assessment Tool (CASSATT). MIT LL has used CASSATT to support several operational needs, including evaluating the improvement in safety when leveraging a SAA system (Kuchar, 2005; Espindle et al., 2009; Kochenderfer et al., 2010c; Griffith and Edwards, 2010; Griffith and Olson, 2011; Edwards, 2012); in the development of standards or trade studies (Griffith et al., 2008; Weibel et al., 2011); and in the development of ACAS X.

The input to a single simulation is the data that describe the encounter situation, including the initial positions, orientations, and nominal maneuvers generated by an encounter model. The simulation’s output is a time history of the aircraft states during the encounter. Running many encounters (e.g., millions) through the simulation enables estimation of the necessary state distributions.

CASSATT is composed of a MATLAB interface and Simulink model where an aircraft is represented by three main pieces: observations, logic and response, and dynamics. This layout is analogous to how TCAS and ACAS X operate. The observation component simulates different sensors and generates an estimate of the intruder’s aircraft position and rates. The logic and response component contains the decision logic and is where the lookup for an optimal policy would be. It also contains a model of pilot response or communication delay.

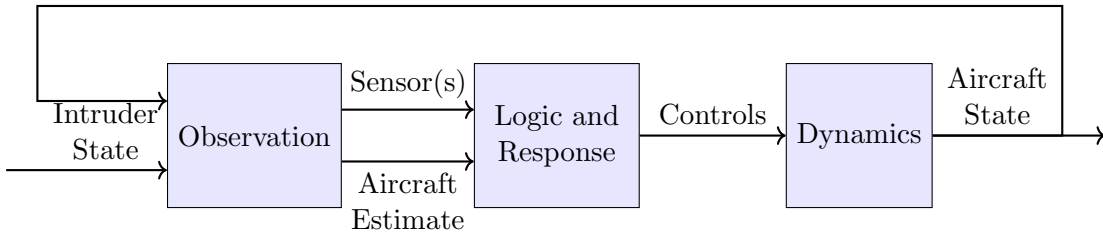


Figure 4-1: Architecture of an aircraft in CASSATT

4.1 Monte Carlo Simulations

The encounter set was sampled from the correlated encounter model, described in Section 2.4, and sampled to be representative of equipped–unequipped encounters in the NAS. Created in January 2013, 500,000 encounters were used for Monte Carlo simulations. Each encounter is weighted with respect to the probability it can occur within the actual NAS. These encounter models capture the behavior of aircraft approximately one minute near CPA.

A perfect sensor was assumed, thus supplying a perfect estimate of the intruder aircraft. These initial simulations implemented no aircraft avoidance logics. The only controls created are those sampled from the encounter model. A simple five second initial delay after the first command and three second delay for subsequent commands are implemented as the pilot model. Finally, these controls are executed via a point-mass dynamics model.

The Monte Carlo simulation was parallelized across 64 processors. 52 states were outputted from the simulation; Appendix A reports these states. The results of each individual simulation was stored in a data file. Each observation is recorded as a floating point number with position-based states recorded with 1.0 precision; rate magnitudes (i.e. vertical rate, airspeed) with 1.0 precision; angular rates with 0.001 precision; and accelerations with 0.001 precision. Each line of the data file contained all the simulation variables for one timestep. Instead of creating one file for each unique simulation, each data file contained up to 10,000 lines and the results for multiple encounters. This 10,000 line limit was anecdotally found to be an efficient for load balancing in latter steps. These raw data files ranged in size from 0.18 Megabytes to 2.56 Megabytes.

A total 13,404 files were produced; assuming 10,000 lines per file, approximately a total of 134,040,000 s were recorded. At 52 states per line, approximately 6,970,080,000 numeric doubles were stored. Safety assessments leveraging CASSATT have traditionally only stored summary statistics for each encounter with approximately up to 13,000,000 numeric doubles comprising the results. By recording every time step, instead of summary statistics, order of magnitudes more simulation data was stored and eventually processed. To the author's knowledge, this is the single largest aircraft avoidance simulation data set ever generated.

4.2 Simulation Processing

Each raw data file is processed to create an associative array used by D4M. An associative array is represented by a triple store of row keys, column keys, and a value. The row key is a concatenation of an individual encounter id and Monte Carlo id. The column key is a concatenation of the simulation time, variable name, and the variable discretized value. Usually a key challenge would be discretizing each variable, however the constant look up time and unlimited column features of D4M significantly reduces this challenge. Since each column requires a constant look up time, the time required to query a range of values grows linearly with the granularity of the feature's discretization. A maximum and minimum range was not required for discretization due to the unlimited columns Accumulo feature. Finally each value in the associative array is one. The associative array is stored in a Matlab .mat file, which range from 0.09 Megabytes to 1.2 Megabytes in size. This step was parallelized using pMATLAB and LLGrid.

4.3 Calculate state-transition matrix

First, a matrix containing a matrix of all state combinations is generated where each column represents a state and a row represents a unique combination of the states. Structure is provided by sorting the rows in ascending order by columns left to right. A kd-tree is generated for each individual state as well. A kd-tree is a binary space partitioning tree often used for range and nearest neighbor searches (Bentley, 1975).

Next, for each action the state-transition matrix is preallocated as an empty sparse matrix. A linear index matrix is also preallocated, which records the linear index for every nonzero element in the state-transition matrix. In Matlab, linear indexing is faster than row / column indexing. While not important for the generate of the state-transition matrix, it is necessary to generate the cost matrix in Section 4.4.

With everything preallocated, a for loop begins and iterates through every associative array i assigned to it. Each unique encounter id in the associative array is identified. A second for loop will iterate over all unique encounter ids j . The elements associated with encounter id j are filtered using D4M and parsed into a matrix. The state variables are then filtered and the kd-tree nearest neighborhood is calculated for each simulation state variable. Using the structure of the sorted state matrix, the state-transition index for each pair of (s, s') is calculated. Leveraging the organized structure is significantly faster with the individual state kd-trees than a single multi-dimensional kd-tree.

For the representative manned aircraft action set, all observed states are considered and used to build the state-transition matrix. For the representative UAS action set with its limited rates, additional filtering was required. Any observation whose vertical rate magnitude is greater than 12.5 ft/s was filtered out. The processing algorithm would recognize time step jumps to prevent introduction of errors from this filtering. Filtering for all states instead of just vertical rate is important because of the dependency of other states on vertical rates, as discovered when building the encounter models (Kochenderfer et al., 2010b).

Finally, the state-transition matrices are updated using the (s, s') linear indices pairs. The value of a nonzero element in S is the sum of the encounter weights for all time steps that each encounter was in that state. Each linear index is recorded and if an NMAC occurred or not at index. The index matrix is then aggregate and the entropy of the state space is calculated.

Depending on the number of dimensions, discretization, and action set calculating the state-transitions on LLGrid requires anywhere from 10 minutes to 2 hours across 64–256 processors. The kd-tree, structured indexing and D4M are the primary drivers that enable the fast processing through the hundreds of millions of samples.

4.4 Generate cost matrix

To generate the cost matrix, first a copy of the associated state-transition matrix is made and assigned as the cost matrix. It is assumed that only the nonzero elements in the state-transition matrix are feasible. Using the nonzero linear indices to generate the cost matrix is required to efficiently manage memory. A naive approach requires too much memory at higher dimensions and does not enable a probabilistic costs. Figure 4.2 illustrates this principle; the blue marks represent the state-transition matrix for a one dimensional state space of vertical separation, Δh . The red lines illustrate what a NMAC only cost matrix would be if all potential NMAC states incurred a cost, regardless of physical feasibility.

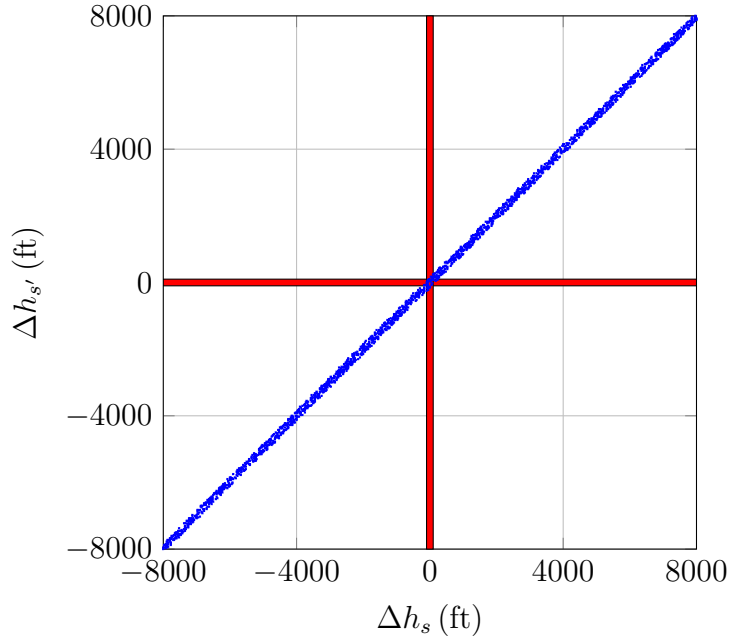


Figure 4.2: $\Delta h(s, s')$ state transitions with naive cost matrix. The blue marks represent the state-transitions and the red bars illustrate a naive NMAC cost matrix.

The overlap between the state-transition matrix and the potential NMAC states is minimal. For example, the transition from $\Delta h_s = -8000$ ft to $\Delta h_{s'} = -8000$ ft isn't

physically possible. Naively assigning cost to potential states drastically increases the quantity of nonzero elements in the cost matrix, needless increasing memory requirements. Furthermore, the naive approach can not support probabilistic cost because it relies on identifying potential states. It would be extremely difficult to identify all potential states using a naive brute force method.

Using the index matrix to identify which elements are associated with an NMAC, the cost functions from Section 3.6 are applied. The alerting region is determined based on the ν_t distribution across states and is calculated as a convex hull of these states in n -dimensional space. Low probability $\nu_t = 1$ states can also be filtered out using a simple standard deviation outliers detection algorithm.

4.5 Optimize policy

With the state-transition and cost matrices generated, an optimal policy is calculated using discounted DP policy-iteration. Optimization required a single desktop and was not parallelized. The algorithm was limited to 100 iterations with a discount value of 0.9 and would iterate until 0.01 epsilon. The algorithm and MATLAB code was provided by the INRA MDP Toolbox (Chadés et al., 2005).

To facilitate a smooth policy, the state matrix was dilated and eroded using a small diamond morphological structuring element. Instead of processing over a hyperrectangle, smoothing was iterated over two free state variables, usually $\{\Delta h, r_h\}$. Smoothing generally modified five percent or less of the space. Convolution over the space with a filter was also considered but modified the structure of the space too much.

4.6 Evaluate policy

The policy is evaluated in CASSATT. As described in Section 3.7, the NMAC risk ratio was used as the safety metric. Figure 4-3 depicts the CASSATT configuration.

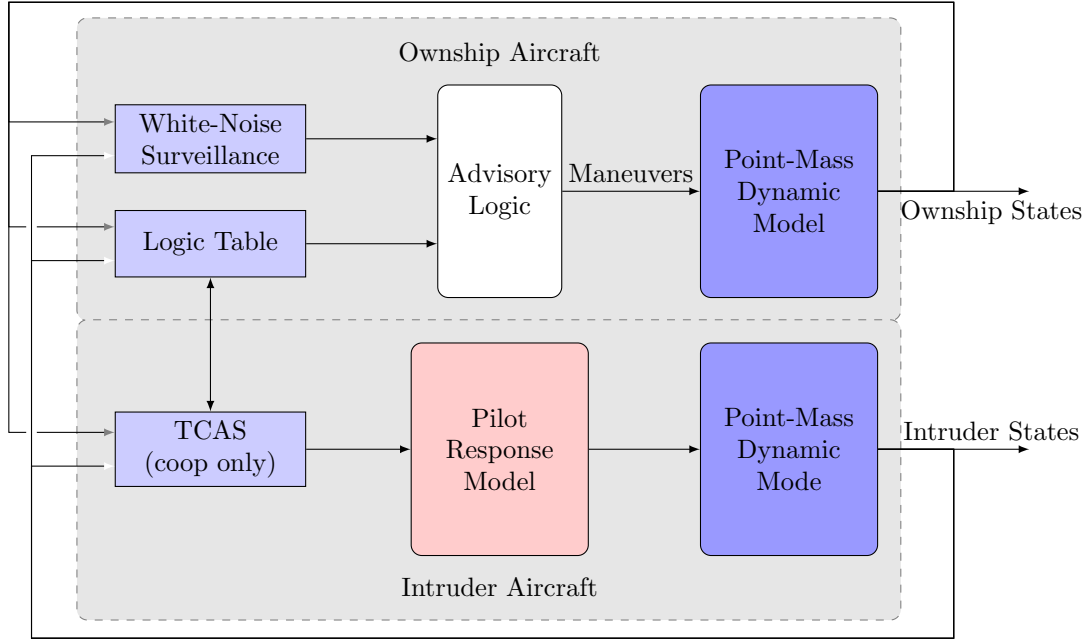


Figure 4-3: CASSATT evaluation framework

A policy will be determined feasible if the risk ratio is 0.1 or less. In addition to evaluating how well the policy mitigates an NMAC, the risk ratio of 5NMAC boundary and average alert length are measured. The 5NMAC is simply a volume five times as large as NMAC as is defined as a loss of separation 500 ft and 2000 ft vertically and horizontally. This metric will quantify how well the policy maintains separation and potential transition to a more dangerous NMAC. The average alert length helps quantifies the operational suitability of the policy. While there is no standard baseline, if assuming a collision avoidance maneuver requires 30 – 60 s to execute under ideal conditions (as determined in Section 3.2), the average alert time should be no more double than that at 60 – 120 s.

Chapter 5

Results and Evaluation

This chapter presents the results of a state space exploration for collision avoidance MDPs via a simulation-based framework. The previous Chapter 4 described the implementation details of this framework. First the ability to generate MDP state-transitions is demonstrated. Following is a discussion of the NMAC entropy for different states and identification of NMAC risk. Using these results, an MDPs was formulated and evaluated.

Discussion emphasizes the trade offs between horizontal and vertical maneuvers, the practicality of specific state spaces, and potential of cost shaping. To facilitate this, most of the analysis considers the optimization as infinite horizon with single decision epoch. The goal is by forcing a single action, clearly defined differences between horizontal and vertical maneuvers can be identified.

5.1 State-transition generation

Section 4.3 describes the process in which state-transition matrices could be generated from any combination of Monte Carlo aircraft simulation data. It is a critical piece of technology development; the research objectives can not be met without this functionality. This capability was verified through analysis of a basic state space containing a well-studied, historical state and comparing the results to historical assumptions. Higher dimensional spaces were then built with a focus on exploring the sparsity and diagonally dominant nature of the space.

First, a one dimensional state space consisting of Δh , a state used by TCAS and ACAS Xa, with a 25 ft/sec discretization was generated:

$$\Delta h = \{-10000, -9975, \dots, 9975, 10000\} \quad (5.1)$$

A convex hull of $\Delta h(s, s')$ is depicted in Figure 5.1. It clearly shows the diagonally dominant, loosely coupled nature of aircraft dynamics and the sparsity of the state-transition matrix. The high degree of sparsity, indicated by the narrowness of the convex hull, is attributed to the one-second transition time step, there is a finite number of states that can be reached in that relative short time step. This is important when considering discrete MDPs, an action must have some probability of transitioning out of the current state or the action will appear to have no effect. Specifically, the sparsity decreases and the convex hull grows wider as discretization becomes more coarse. However, the matrices remain diagonally dominant for all but the most coarse discretizations. These general trends were observed for both the manned and UAS actions sets.

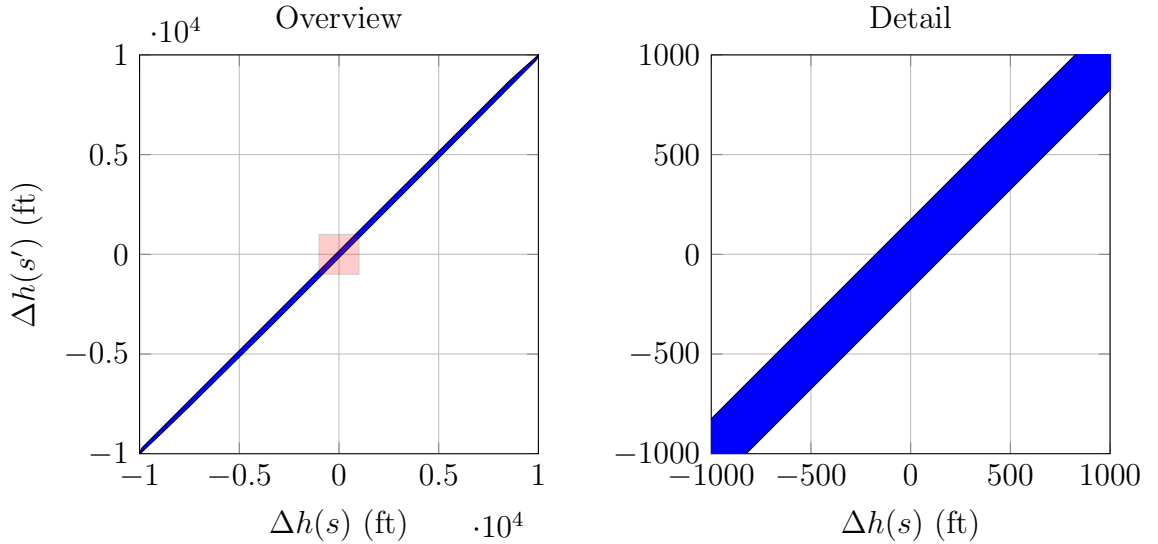


Figure 5.1: Convex hull of $\Delta h(s, s')$ using manned action set

While Figure 5.1 illustrates a general trend but a more detailed analysis is required in addition. Through examining of the lightly Gaussian smoothed probability density function (PDF) of $\Delta h(s)$ in Figure 5.2, specific conclusions concerning about each action are made.

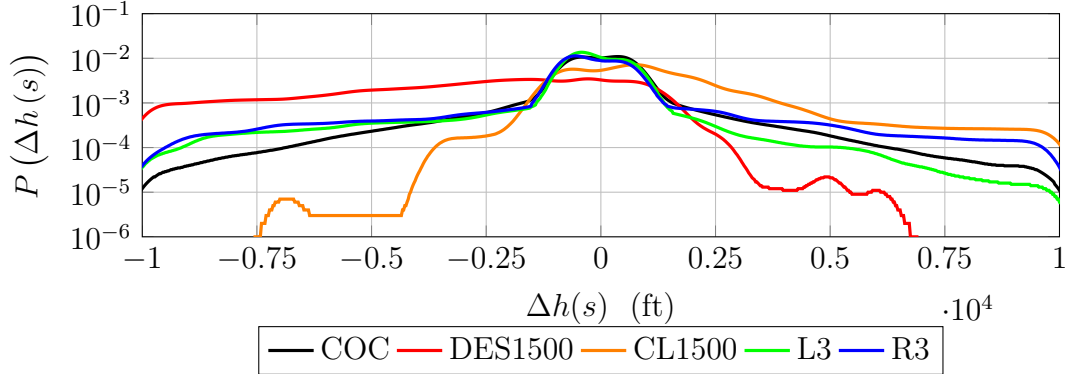


Figure 5.2: PDF of $\Delta h(s)$ using manned action set

As expected, there is a greater correlation between vertical maneuvers and Δh than horizontal maneuvers and Δh . The distribution is also sensitive to the sense of the vertical maneuver. If the aircraft is descending, the PDF indicates a higher probability of the controlled aircraft would be below the intruder. A similar trend exists from climbing and a positive Δh . Simply, an descending or climbing aircraft should eventually transition below/above an intruder and influence Δh more than horizontal maneuvers. Additionally, Δh is independent of the sense (left or right) of the horizontal maneuver. Since turning primarily influences the horizontal plane, this is expected since Δh is part of the vertical axis.

Additional state spaces were then analyzed for sparsity and expected behavior. Table 5.1 reports the MATLAB memory requirements for representing random state-transition matrices. Substantial memory savings are realized with a sparse representation. For large matrices, such as a $150,176 \times 150,176$ with 22,552,830,976 elements, a sparse representation is required; a dense representation is not possible.

Table 5.1: Memory requirements for random state-transition matrices.

Matrix Size	Nonzero Percentage	Memory (Megabytes)		Memory Saved
		Dense	Sparse	
181×181	100%	0.2621	0.2038	22.26%
608×608	6.2749%	2.9573	0.376	87.29%
641×641	1.818%	3.287	0.1247	96.2%
820×820	5.1911%	5.3792	0.565	89.5%
$73,568 \times 73,568$	0.0545%	43,298.005	47.7434	99.89%
$150,176 \times 150,176$	0.019%	N/A	69.9005	∞

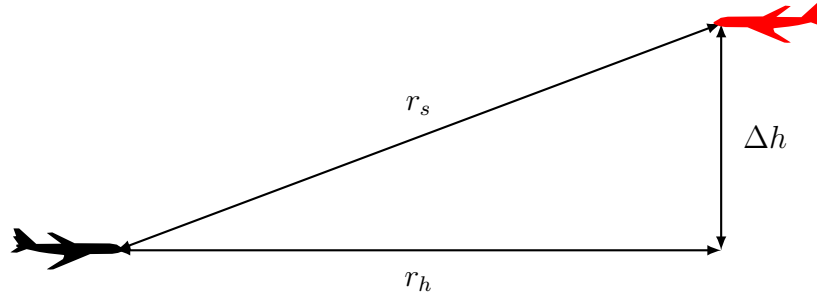
These results provides a common sense sanity check of the framework and verify the capability to generate appropriate state-transition matrices. Using this capability, the NMAC entropy for each potential state can be calculated.

5.2 State Entropy

States are categorized by their function in collision avoidance and reports NMAC entropy and memory requirements. Based on this, different state spaces composed of memory efficient and high entropy states are generated. Note that this section's figures leverage the manned action sets but the conclusions are similar between sets.

5.2.1 Separation

A state space needs to represents the separation (distance) between the aircraft. This information corresponds to the position of the aircraft in space. As discussed in Section 3.5, there are three different coordinate systems commonly used. The Cartesian and Cylindrical systems use the same state to quantify the vertical axis with the relative vertical position between two points as Δh . In the Cylindrical system, r_h is the relative horizontal position calculated as the euclidean distance. The Spherical system has a single relative position, and is the radial distance r_s . For each of these three separation states, eight different uniform discretizations were generated. Figure 5-4 reports the corresponding NMAC entropies and memory requirements.

Figure 5.3: Separation states: $\Delta h, r_h, r_s$

It is not surprising that r_s and r_h have similar NMAC entropy and yield greater NMAC entropy than Δh . Since Δh is often orders of magnitude smaller than horizontal separation, the vertical component of r_s doesn't provide much more information. Since Δh is primarily influenced by only four variables $(h_i, h_o, \dot{h}_o, \dot{h}_i)$ and r_h is influ-

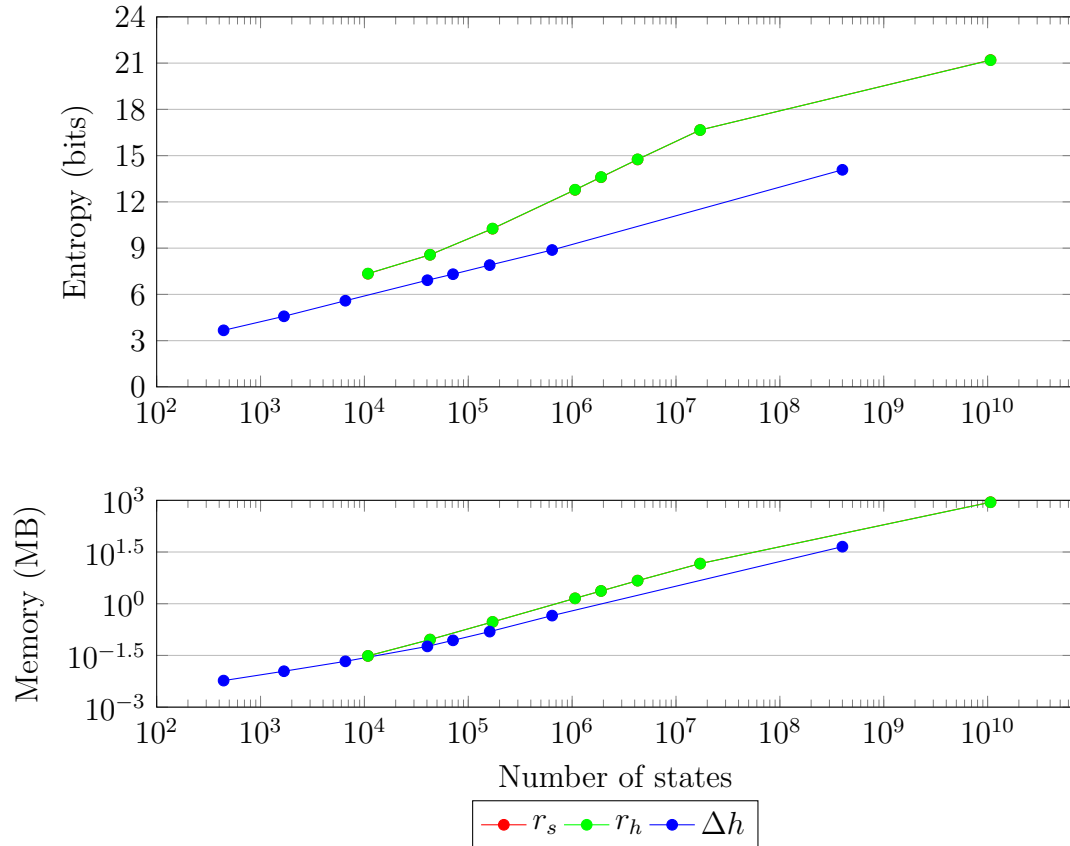


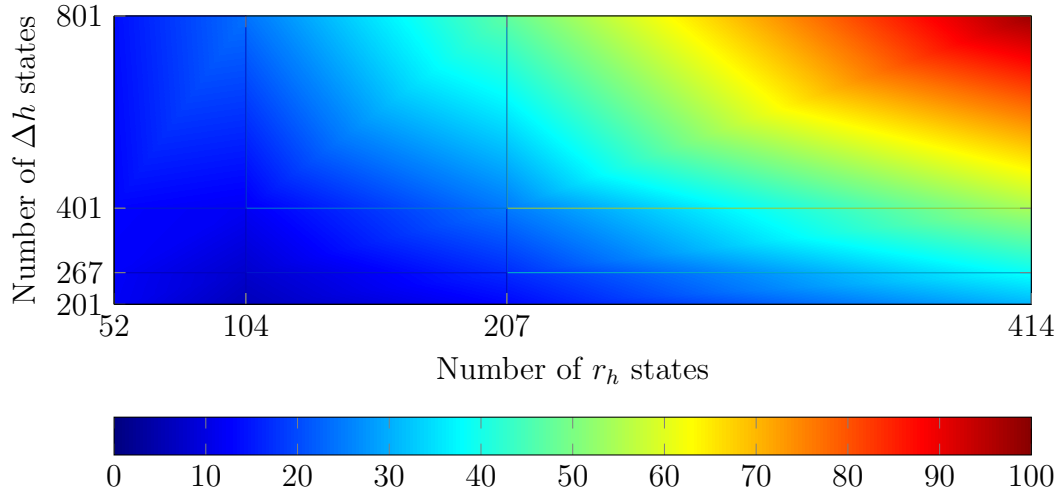
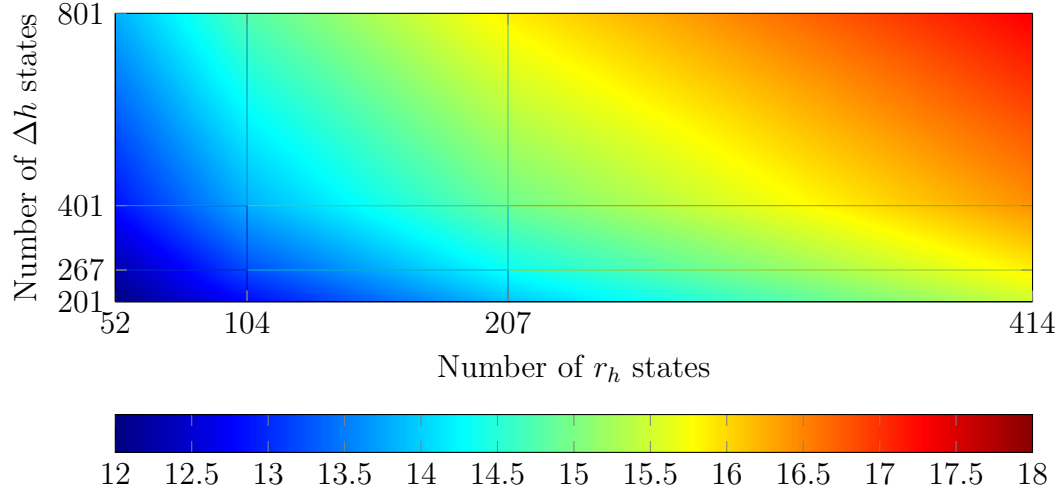
Figure 5.4: Entropy and memory of individual COC range states

enced by many more variables, $(n_o, n_i, e_o, e_i, v_o, v_i, \psi_o, \psi_i)$, it is reasonable to expect that r_h should have greater entropy. Furthermore, r_h will have greater mutual information with many more states than Δh .

Although r_s and r_h have significant NMAC entropy, very discretized uniform representations are not feasible due to their memory requirements. A r_s state-transition with 10,669,857,025 elements requires 882 MB. From a memory perspective, r_s and r_h are more efficient. A Δh state-transition matrix 400,040,001 elements requires 45 MB and has an NMAC entropy of 14.08 bits. Yet a r_h state-transition with 1,898,884 elements requires only 2.3412 MB and has an NMAC entropy of 13.61 bits. This efficiency is attributed to the scale of the horizontal plane, it maybe more beneficial to quantify if aircraft are a nautical mile apart horizontally than a thousand feet vertically.

Expanding upon this, 16 different uniform discretizations of $\{r_h, \Delta h\}$ state space were generated and analyzed; Figure 5.5 illustrates the two dimensional NMAC entropies and memory requirements. Both metrics have wider range along r_h than Δh . It provides an insight into the how the discretization of the different states correspond to NMAC and helps addresses the question, “Is it better to finely discretize r_h or Δh ?” While adding more states to r_h results in greater entropy on a per state basis, it also increases memory at a faster rate than Δh . Conversely, adding additional states to Δh increasingly has diminishing returns. This is attributed to common flight trajectories, once aircraft reach a cruising altitude, there are minimal altitude changes.

Finally with negligible differences between r_s and r_h , it was decided to leverage r_h in analysis because it persevered the assumption of independent horizontal and vertical components, described in Section 3.5.



5.2.2 Range Rates

The rates in which the separation states change are discussed next: $\dot{\Delta h}, \dot{r}_h, \dot{r}_s$. Figure 5.7 shows that similar to the ranges, rates that include horizontal information have greater NMAC entropy and memory requirements. Interestingly, course uniform discretizations of $\dot{\Delta h}$ produce little NMAC entropy and at even fine discretizations, such as $\dot{\Delta h} = \{-3000, -2999, \dots, 2999, 3000\}$, yield significant less than \dot{r}_h or \dot{r}_s .

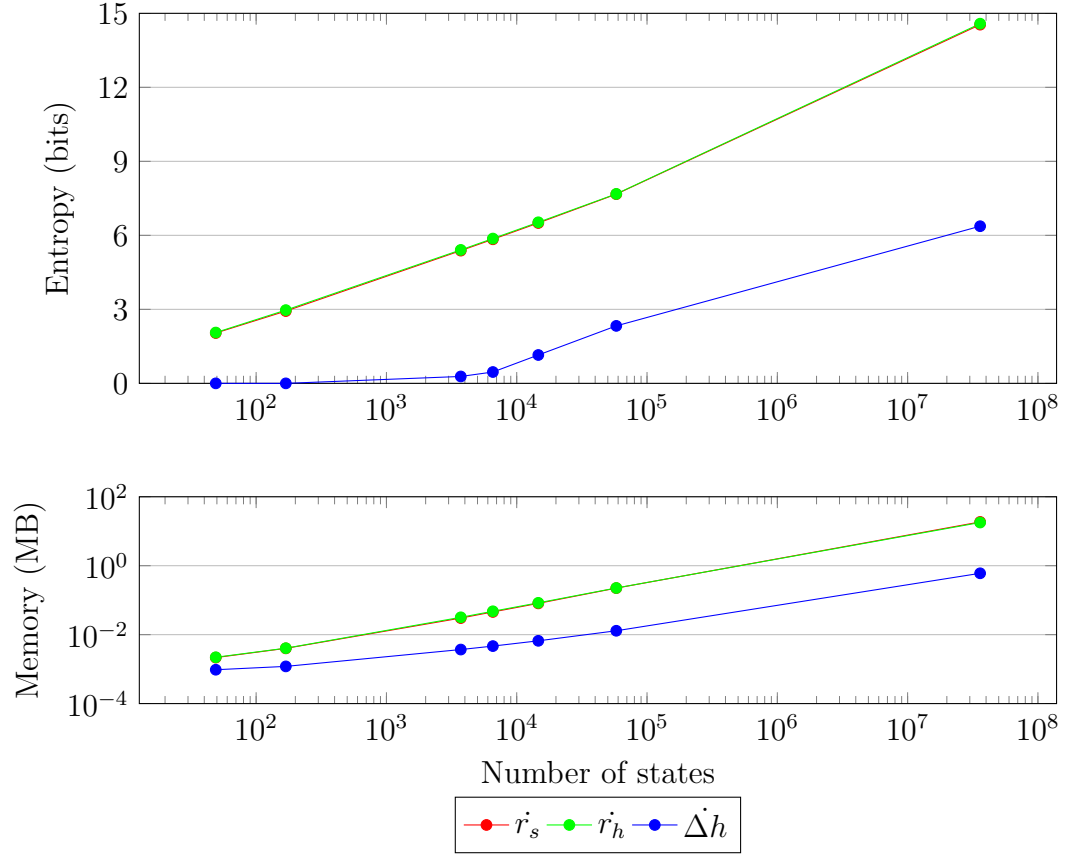


Figure 5.7: Entropy and memory of range rate states

5.2.3 Simple Tau

The separation and range rate states can be aggregated together to decompose either the horizontal plane or vertical axis into a τ state. While TCAS implements a slightly more complex modified τ and ACAS X defines it time as loss of separation, this analysis uses the simple definition of τ :

$$\tau_v = \frac{\Delta h}{\dot{\Delta h}} \quad (5.2)$$

$$\tau_h = \frac{r_h}{\dot{r}_h} \quad (5.3)$$

$$\tau_s = \frac{r_s}{\dot{r}_s} \quad (5.4)$$

TCAS has discretized τ_h from 0–45 s but to facilitate state space exploration, two different τ ranges were considered: 0–50 s and 0–100 s. Similar to the separation and range rate results, the NMAC entropy for the horizontal and spherical τ in Figure 5-8 have negligible differences and yield greater NMAC entropy than τ_v . These states are not sparse, there are no nonzero elements in the state-transitions. Thus, the memory required is a linear function of the discretization and range.

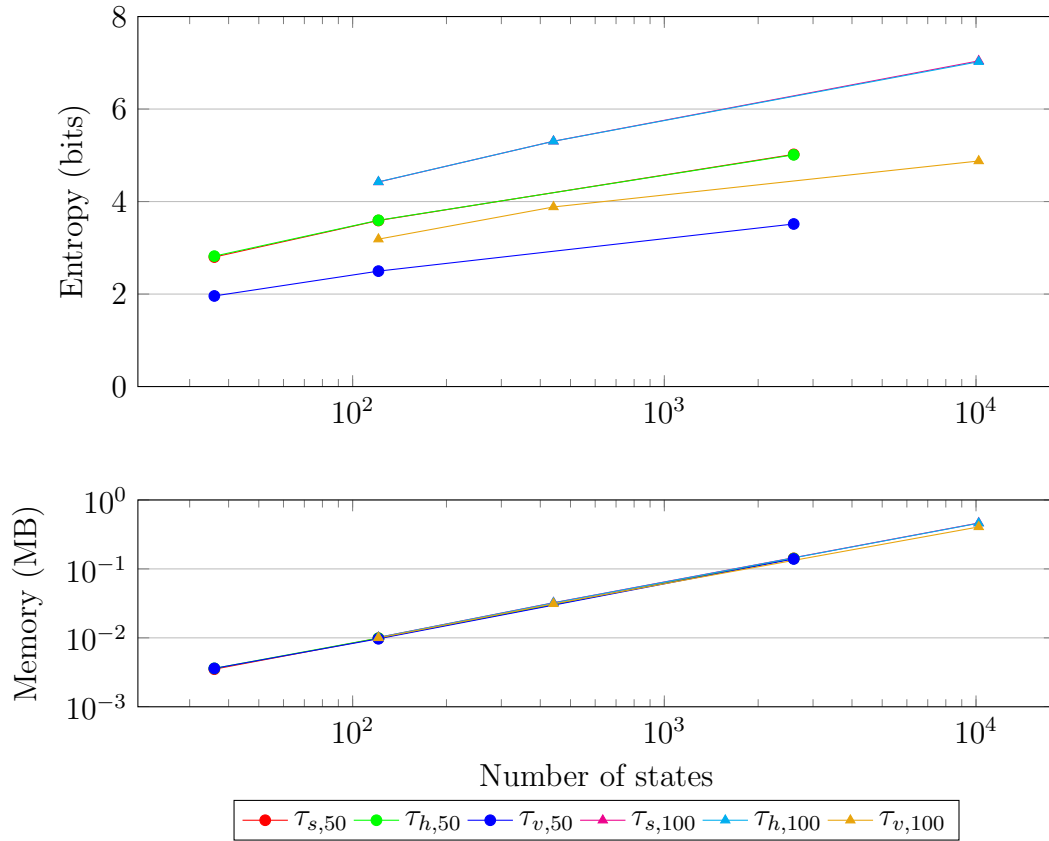


Figure 5-8: Entropy and memory of COC simple τ states

For all three τ states, increasing the range out to 100 s led to greater NMAC entropy. More importantly, a coarse discretization with a maximum of 100 s yielded greater NMAC entropy than a finely discretized state with a maximum of 50 s given a constant memory requirement.

Vertical and horizontal maneuvers also influence τ differently, as illustrated by Fig-

ure 5.9. Saturation, as described in Section 3.5.3, near the maximum is also clearly evident by the significant increase in likelihood near the maximum extreme. The saturation is greater when using a maximum of 50 s. For τ_h , vertical maneuvers produce greater NMAC entropy than horizontal maneuvers. Since horizontal maneuvers can easily turn into or turn away from intruders, τ is significantly more variable and provides less information about a potential NMAC. Conversely, vertical maneuvers can more easily lead to a lost of separation and a potential transition into an NMAC.

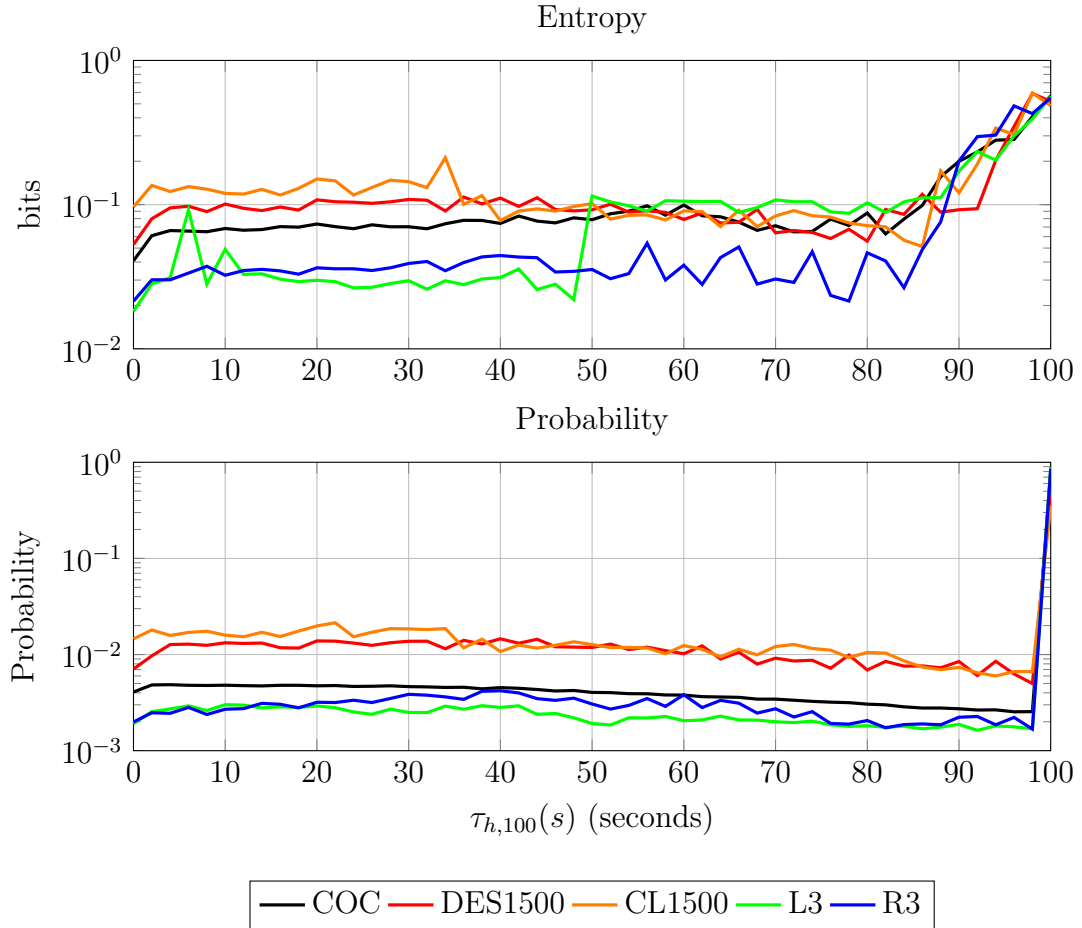


Figure 5.9: NMAC entropy and probability distribution for $\tau_{h,100}(s)$

Examining specific $\tau_{h,100}(s)$ values, in Figure 5.10, highlights the variability of

τ states and the analytic benefits of NMAC entropy. The x-axis is the transition state $\tau_{h,100}(s')$ that is potentially reached by a specific $\tau_{h,100}(s)$. For each $\tau_{h,100}(s)$ the probability to transitioning to neighboring values is Gaussian. As $\tau_{h,100}(s)$ decreases, the tails of the distribution grow and the peak shrinks. Similar behavior was observed for τ_s and τ_v states and for ranges of 0–50 s.

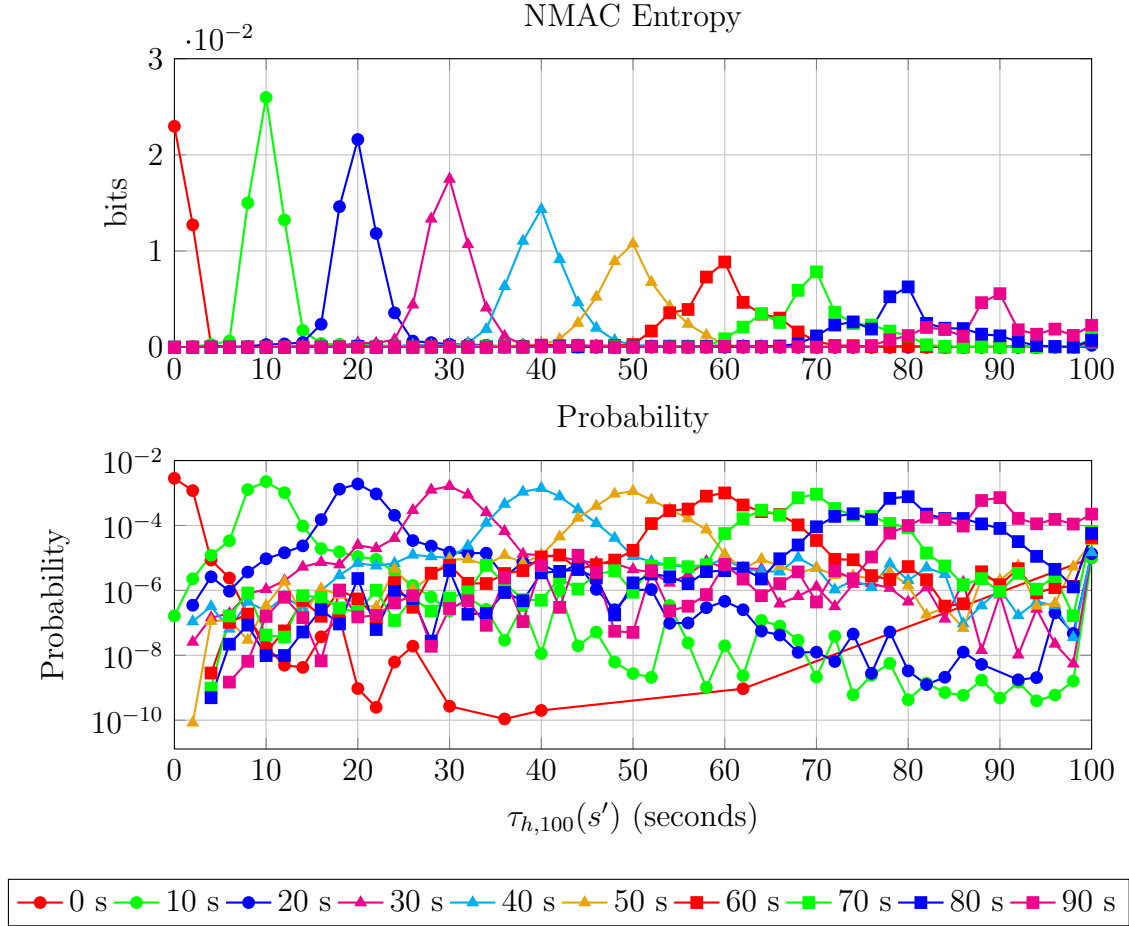


Figure 5.10: Entropy and probability distribution for COC $\tau_{h,100}(s)$

From an optimization perspective, there is more information about the potential state transition for a low value $\tau_{h,100}(s)$ state than high value state. A higher value state, such as $\tau_{h,100}(s) = 90$ can rapidly transition to a more dangerous low value state; closely-spaced parallel encounters are a type of encounter that can cause this.

This is a consequence of τ being an approximation that address a lack of azimuth information due to the limitations of surveillance technology in the latter half of the twentieth century. This NMAC entropy perspective does not devalue the utility of a τ state but rather quantifies the amount of information that each τ state provides about a potential transition to an NMAC or a different state.

5.2.4 Angles

The separation states do not completely quantify the encounter geometry, the angle between the aircraft is extremely useful. TCAS and ACAS Xa do not require this information because they only operate in the vertical axis while ACAS Xu includes angular states. The the ratio between headings ($\Delta\psi$), the spherical inclination angle (θ_s), and the horizontal resultant angle (ψ_r) were evaluated. A combinational challenge is reducing the number of state-transitions, discretizing from -180 deg to 180 deg with 1 deg intervals, produces a state-transition matrix with 130,321 nonzero elements, which when combined with other states can easily grow to a large state space. Finally, the angular rates are an intriguing category due to the historically poor azimuth observations associated with aviation surveillance. It has been historically very difficult to provide an accurate angular measurement, let alone the derivative of them to calculate the angular rate changes

The aggregated feature ψ_r yielded greater entropy than the traditional relative heading state $\Delta\psi$. Compared to other states, exponentially increasing in the number of states does not correspond with significant increases in NMAC entropy. These angular states, along with τ states, were not sparse. The memory requirements between angular states vary due to the state ranges, not all span 360 deg. For a state such as $\Delta\dot{\psi}$, the amount of entropy can be approximately doubled with approximately a 500% increase in memory.

Similar to the separation states, the vertical state angle θ_s yielded the least NMAC

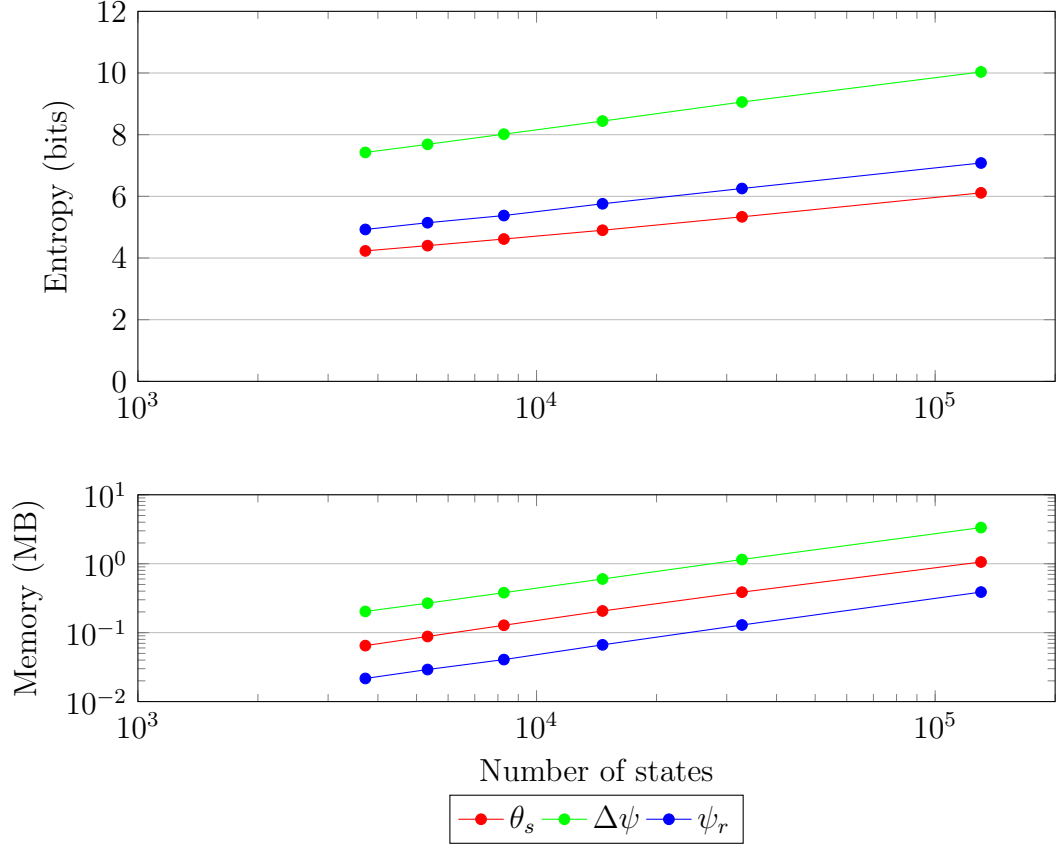


Figure 5.11: Entropy and memory of COC angle states

entropy. Δh also provides more entropy than θ_s on a per memory basis, indicating that Δ is more efficient in quantifying the vertical axis. This highlights the disadvantages of the spherical coordinate system. The mathematical intuition discussed in Section 3.5.2 correlates with the NMAC entropy result. Figure 5.12 depicts that as r_s increases, the difference between θ_s states for different Δh states becomes small. Compared to all states, θ_s yielded some the lowest NMAC entropy overall.

5.2.5 Airspeed

The airspeed states consists of the ownship airspeed (v_o), intruder airspeed (v_i), the resultant vector magnitude (\vec{R}), and a collection of aggregate features generated from (v_o, v_i) . Figure 5.13 foremost illustrates that the aggregating operation (summation,

Figure 5.12: Spherical perspective for a constant Δh

distraction, etc.) had little effect. The \vec{R} state is the most complicated and aggregates many states, including the heading information of both aircraft. This angular information is a reason why \vec{R} yields the greatest NMAC entropy but also why \vec{R} requires memory. However, complexity and aggregated features do not guarantee good NMAC entropy results, as shown by the modified ratio state $(v_o/v_i)^{*1}$, which yielded the least NMAC entropy of the airspeed states.

The lack of differences in the results indicate the challenge of identifying similar features of airspeed states. Two-dimensional spaces of $\{r_h, v_o\}$ and $\{r_h, v_o + v_i\}$ were generated and analyzed to quantify the potential of aggregate airspeed states. $\{r_h, v_o + v_i\}$ was more sparse and required less memory than $\{r_h, v_o\}$. NMAC entropy of these two-dimensional state spaces ranged from approximately 11–16 bits. The aggregate feature state generally required 10–15% less memory, while providing 2–5% more NMAC entropy.

5.2.6 Vertical Rates

The vertical rate states consist of ownship's vertical rate (\dot{h}_o), intruder's vertical rate (\dot{h}_i), and a collection of aggregate features generated from (\dot{h}_o, \dot{h}_i) . Illustrated by Figure 5.14, the simple aggregated feature of summing the vertical rates, $(\dot{h}_o + \dot{h}_i)$

¹ $(v_o/v_i)^* = \mathbb{I}_{(v_o/v_i) \geq 1} (v_o/v_i) + \mathbb{I}_{(v_o/v_i) < 1} (-v_i/v_o)$

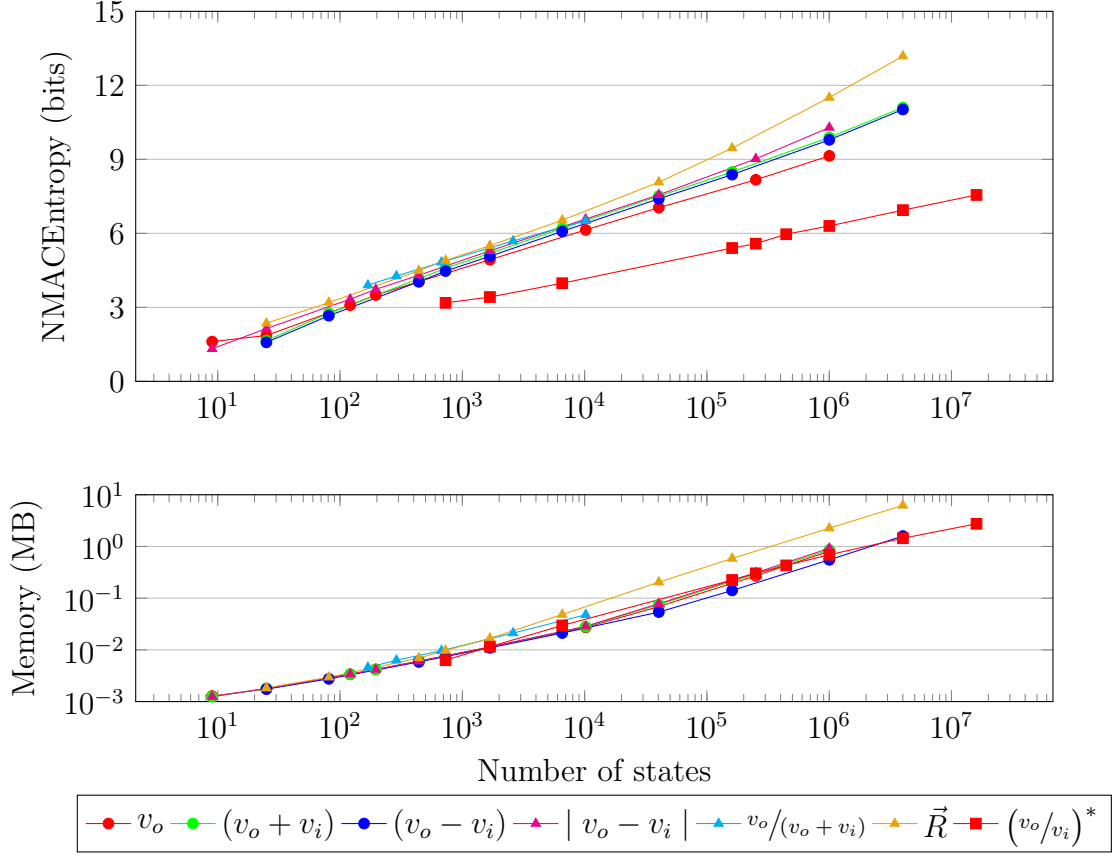


Figure 5.13: Entropy and memory of COC airspeed states

yielded the greatest NMAC entropy with less memory at most discretizations. These characteristics identified it as a promising candidate to be included in the complete state space. However, unlike the other categorizes, the ranks of highest to lowest NMAC entropy, but not memory, varied across states. For example, \dot{h}_o/\dot{h}_i yielded more NMAC entropy than $|\dot{h}_o| + |\dot{h}_i|$ at smaller state discretizations, but the reverse occurs at finer discretizations. Yet \dot{h}_o/\dot{h}_i always requires more memory.

Figure 5.15 shows the NMAC entropy and probability distributions for specific $(\dot{h}_o + \dot{h}_i)$ values. This state is very symmetric. There is significantly greater NMAC entropy at $(\dot{h}_o + \dot{h}_i) = 0$ and greater risk of NMAC when neither aircraft are maneuvering vertically. This is expected, if two aircraft are within ± 100 ft and the vertical rate remains unchanged, then the risk NMAC will be relatively high. More impor-

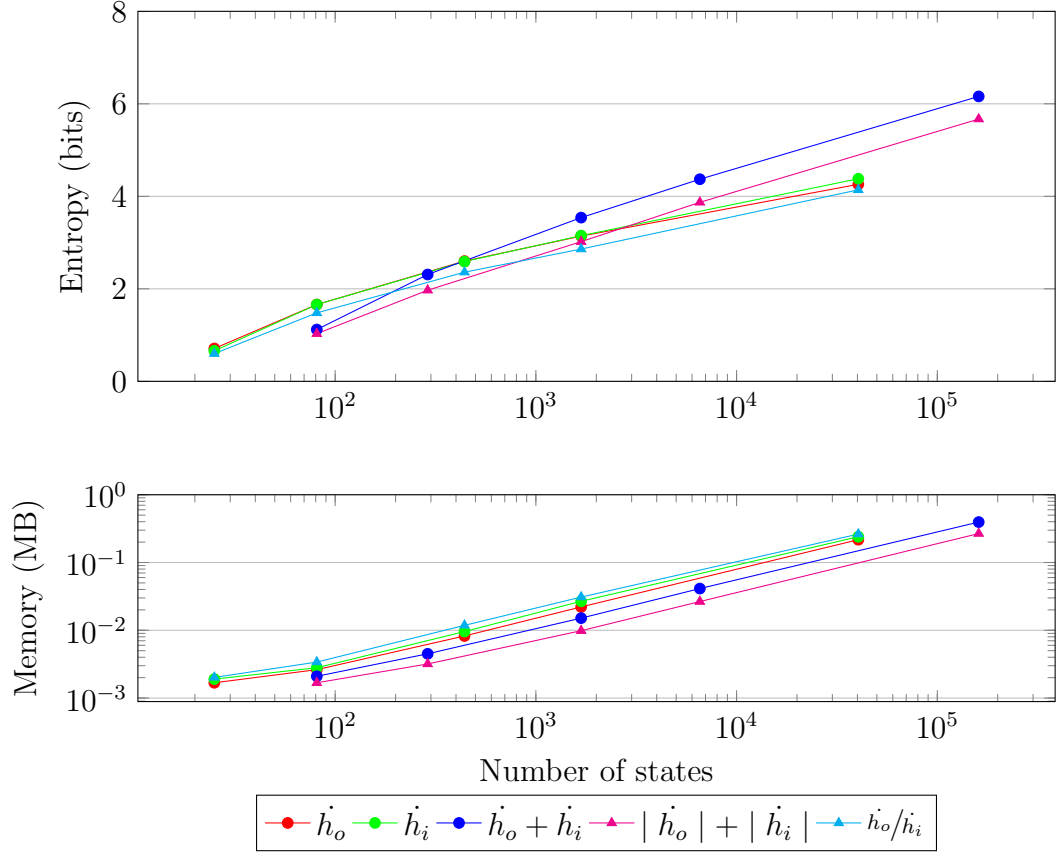


Figure 5-14: Entropy and memory of COC vertical rate states

tantly, this state accounts for different combinations of vertical rates. Specifically if both aircraft are in level flight but also accounts for when aircraft have opposite vertical senses. As discussed in Section 3.5.2, if a state space included $(\dot{h}_o + \dot{h}_i)$ and \dot{h}_o , then it is possible to distinguish between the different vertical rate combinations.

The benefits of aggregation are persevered when expanding to larger state spaces. The NMAC entropy and memory requirements for 16 different discretizations of each $\{\Delta h, \dot{h}_o\}$ and $\{\Delta h, \dot{h}_o + \dot{h}_i\}$. The memory requirements, given a number of states, is nominally similar between the different state spaces. However, the NMAC entropy degradation as the number of states decreases is different. $\{\Delta h, \dot{h}_o + \dot{h}_i\}$ space perseveres more NMAC entropy as the discretizations become more course and on

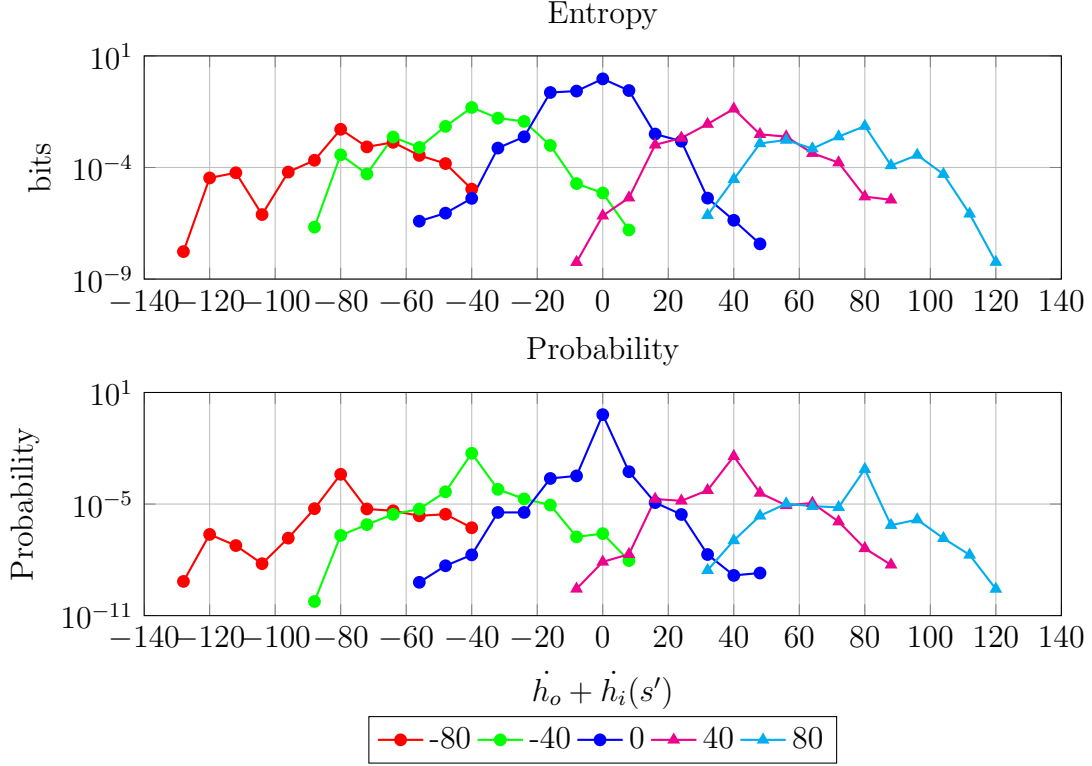


Figure 5-15: Entropy and probability distribution for COC $\left(\dot{h}_o + \dot{h}_i\right)$

average provide greater NMAC entropy. This indicates that $\left\{\Delta h, \dot{h}_o + \dot{h}_i\right\}$ is more computationally efficient than $\left\{\Delta h, \dot{h}_o\right\}$.

5.3 Identifying NMAC Risk

The NMAC entropy of a state provides a one-step perspective of the state's utility. Calculating and analyzing the NMAC horizon (ν_t) provides a longer multi-step perspective with regards to safety. Both these metrics will then be used to formulate a MDP and generate a policy in Section 5.4.

While the simulation framework can support any time horizon, this analysis considers horizons of 15 s, 30 s, and 45 s. Starting with a baseline space of $\{r_h, \Delta h\}$, additional states are added and evaluated for how the additional states change the

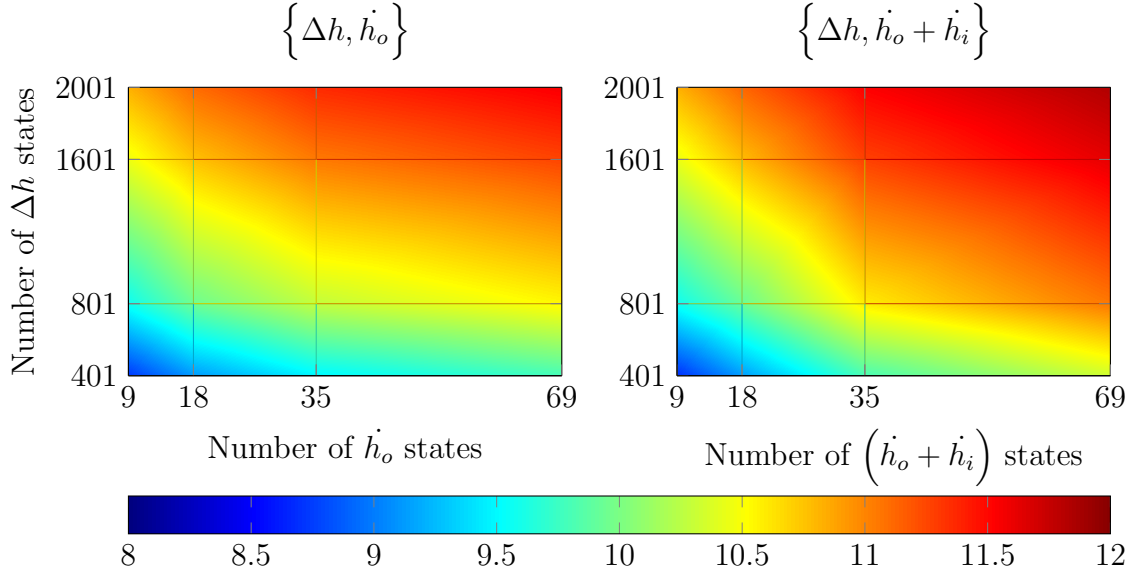


Figure 5-16: Entropy (bits) of COC $\{\Delta h, \dot{h}_o\}$ and $\{\Delta h, \dot{h}_o + \dot{h}_i\}$

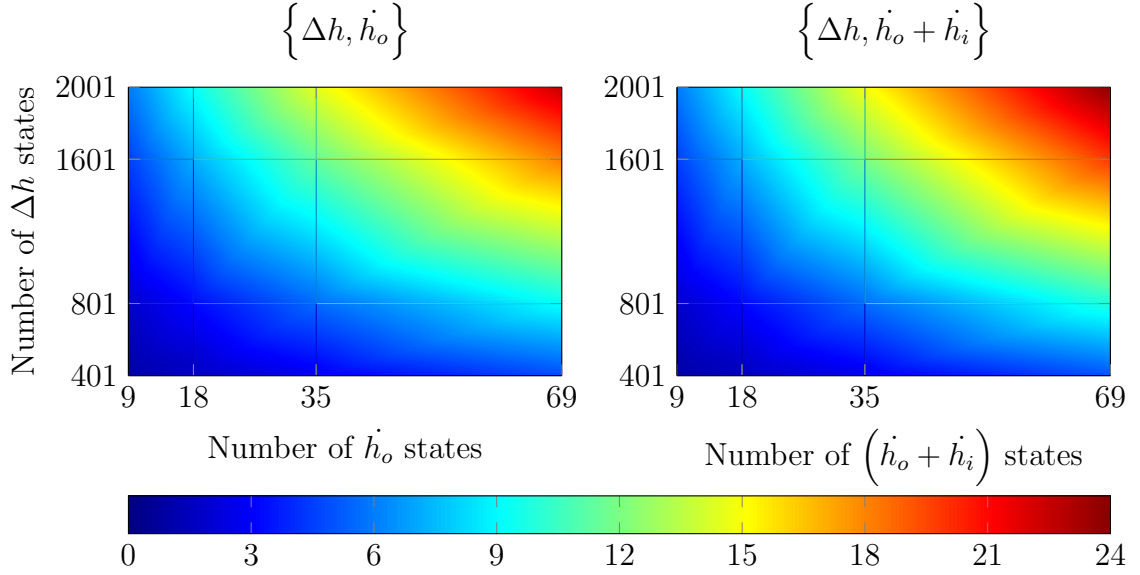


Figure 5-17: Memory (MB) of COC $\{\Delta h, \dot{h}_o\}$ and $\{\Delta h, \dot{h}_o + \dot{h}_i\}$

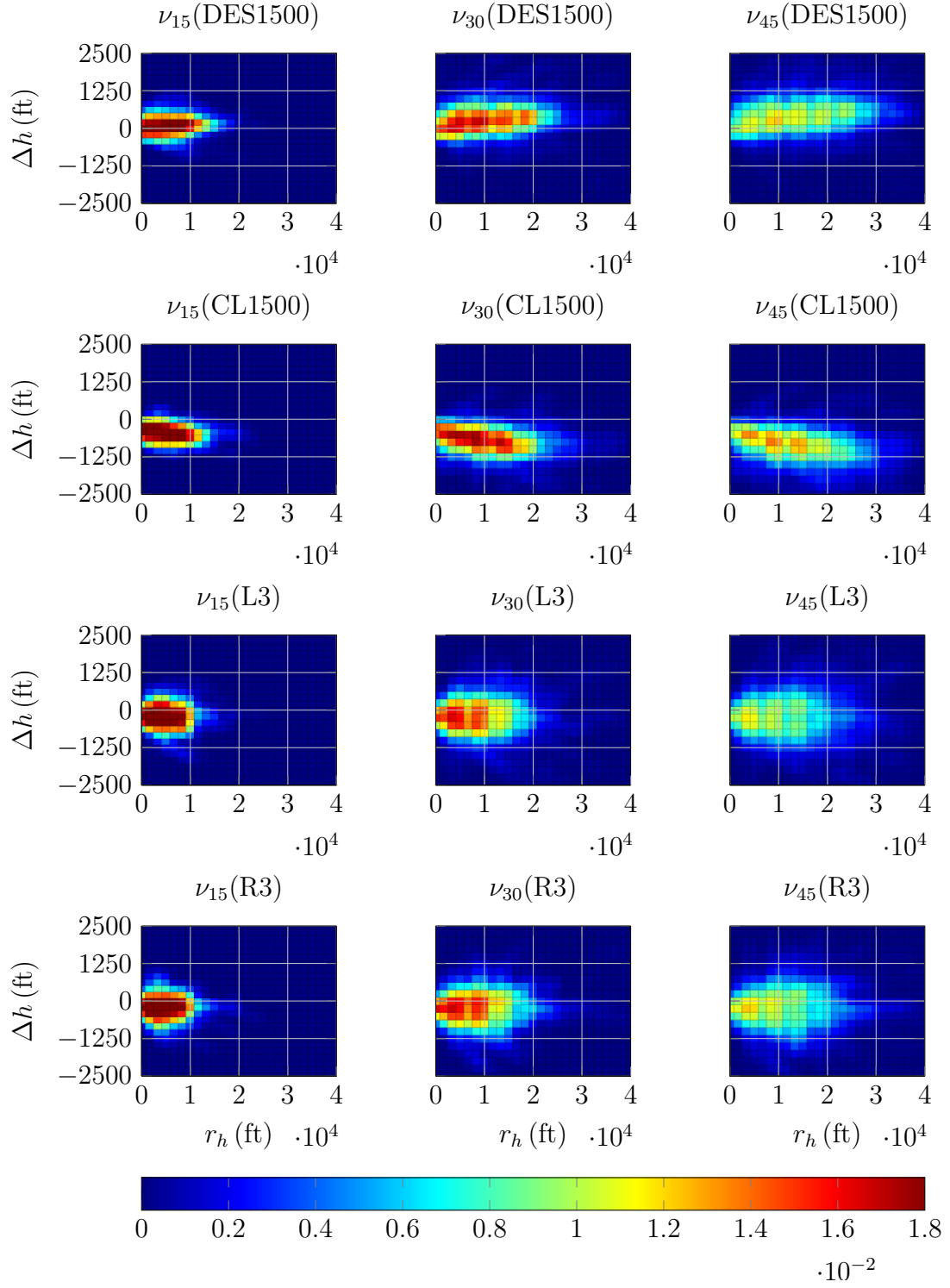
ν_t distribution. A convex hull of the NMAC risk is calculated. Instead of calculating other ν_t distributions, the resulting convex hull can be influenced through filtering. The default filter requires each state to have at least two observations where $\nu_t = 1$

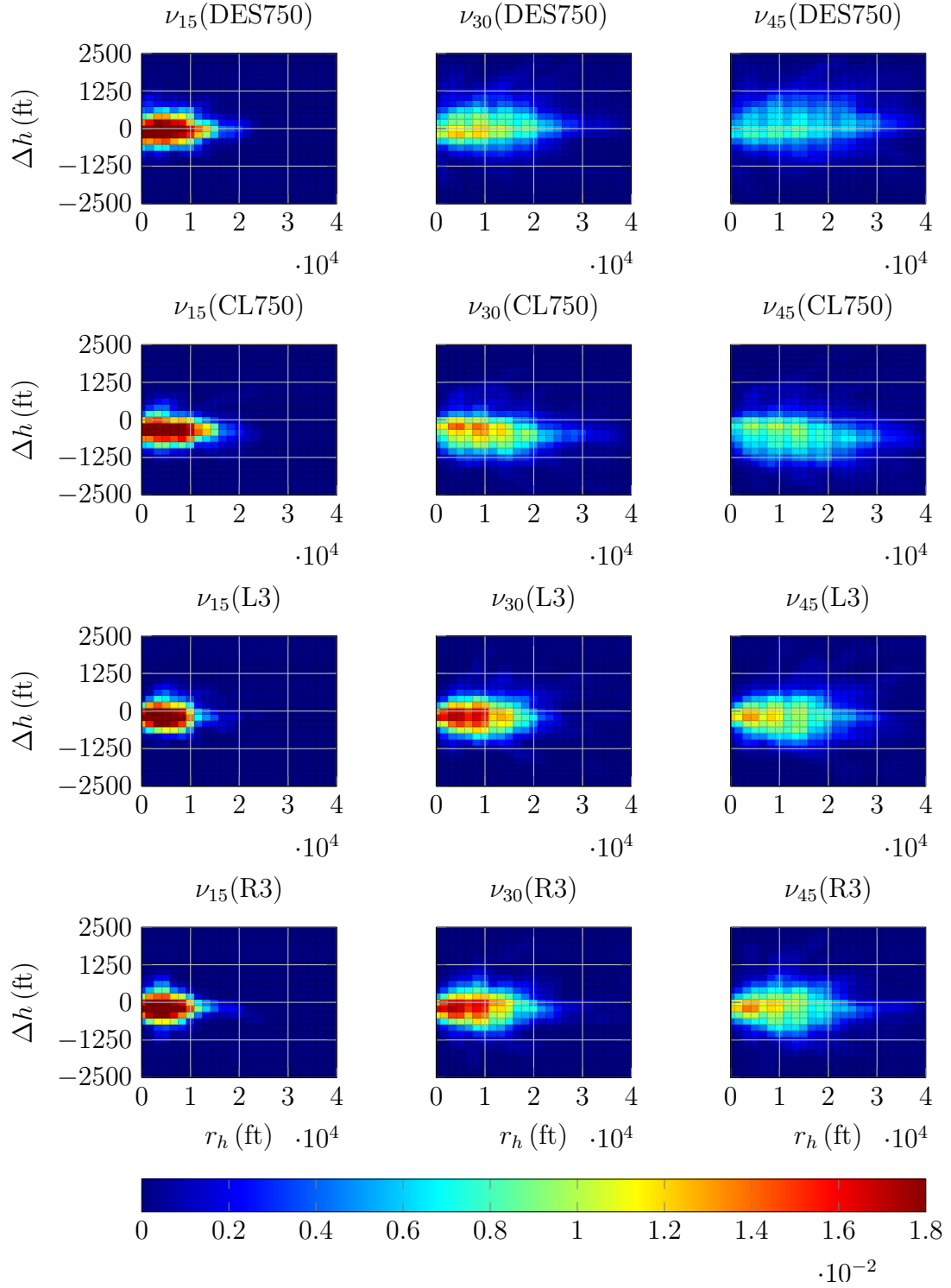
and implements a basic standard deviation outlier filter. Changing the filtering parameters will change the shape of the convex hull and subsequent alerting region of the MDP formulation.

5.3.1 Baseline $\{r_h, \Delta h\}$ State Space

Figure 5-18 shows the normalized distribution for the actions over r_h and Δh for three different NMAC horizons: ν_{15} , ν_{30} , ν_{45} using the representative manned aircraft action set; Figure 5-19 uses the representative UAS action set. For all actions, the NMAC risk becomes more diffuse across the space as the time horizon increases, the NMAC risk for any nonzero ν_t state increases as t decreases. This is expected since with a larger t , there is more time for aircraft to maneuver away from each other. Weibel (Weibel et al., 2011) observed a similar trend when defining risk as a function of τ . For all results, at least 80% of the NMAC risk resides when $r_h \leq 15,000$ ft, which is approximately the maximum range at which TCAS may issue an RA.

Although the different actions share some common conclusions, they also significantly influence NMAC risk. For horizontal maneuvers, majority of the risk is concentrated when $r_h \leq 1000$ ft and is skewed much closer to $r_h = 0$ than for vertical maneuvers. When $r_h > 20,000$ ft, vertical maneuvers are generally subject to more risk too.

Figure 5-18: NMAC horizons using manned action set for $\{r_h, \Delta h\}$

Figure 5-19: NMAC horizons using UAS action set for $\{r_h, \Delta h\}$

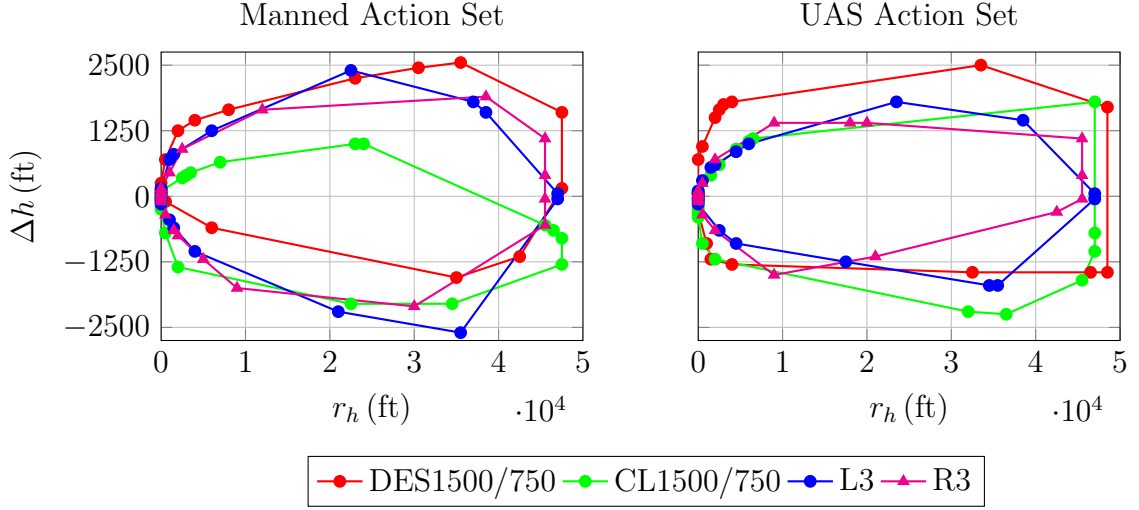
Specifically, if Δh is negative and an ownship is descending or if Δh is positive and ownship is climbing, it is very unlikely to transition to an NMAC. This is as expected because separation will only decrease if the intruder has a vertical rate the same sense and a greater magnitude. Both horizontal action have approximately symmetric distributions along the vertical axis, unsurprising because turn rate does not directly influence Δh . Figure 5-20 illustrates this.



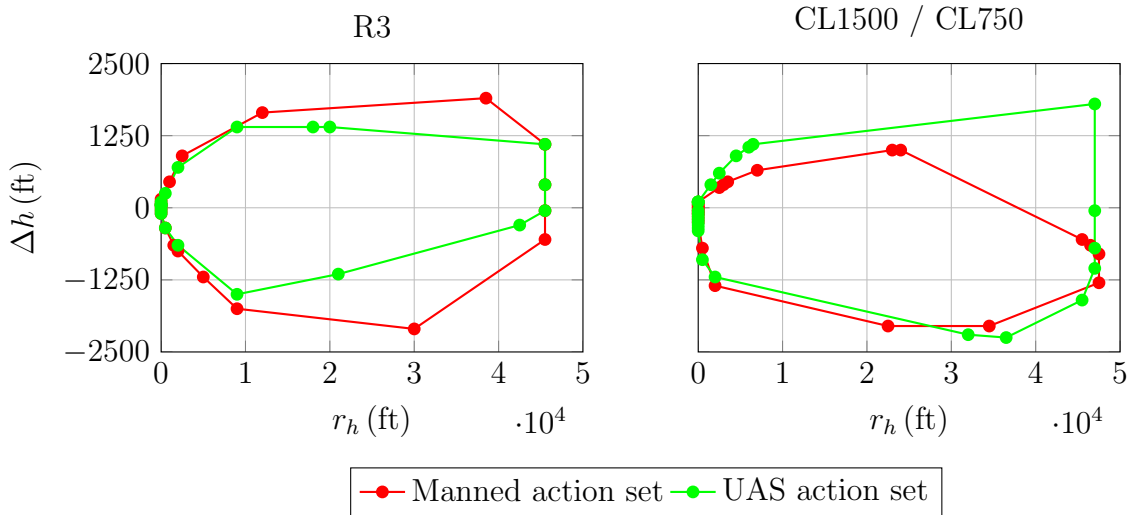
Figure 5-20: Vertical perspective within regards to action

The NMAC risk distribution is also dependent upon the action sets. Figure 5-21 illustrates the convex hulls of ν_{30} of each action for both action sets. The convex hulls include any nonzero ν_{30} value and does not take into account the probability of the nonzero ν_{30} element. The hulls of the different actions overlap with each other and the representative UAS hulls are similar to each other than.

When directly comparing the action sets in Figure 5-22, it is clearly evident that vertical performance constraints directly influence NMAC risk. Horizontal actions, such as R3, contract ν_{30} contour along Δh . Since the vertical rate performance was limited with the UAS action set, $|\dot{\Delta h}|$ on average is smaller because $|\dot{\Delta h}_o|$ on average is also smaller. Larger Δh states are less likely to transition to a riskier ν_{30} state, thus the contraction. Interestingly, when unioning the action hulls, there is marginal difference between the action sets if the probability of ν_t is ignored.

Figure 5.21: Comparison of $\nu_{30} \{r_h, \Delta h\}$ convex hulls

Conversely, the ν_{30} distributions expands along Δh when using the UAS action set. For example, if ownship climbing when $\Delta h \geq 0$, that is no longer sufficient to minimize NMAC risk, especially at larger r_h states. This is due to that the magnitude of the intruder's vertical rate is usually greater than the UAS ownship's. The ν_{30} distribution of the UAS action set is even wider than the ν_{45} of the manned action set. As a consequence, the UAS action set will produce a larger alerting volume.

Figure 5.22: Comparison of specific action $\nu_{30} \{r_h, \Delta h\}$ convex hulls

5.3.2 Adding States

A state space of $\{r_h, \Delta h\}$ isn't sufficient because it doesn't quantify the rate at which the encounter system is moving and lacks an angular component to completely describe the geometry of the encounter. Without this additional information, the DP algorithm will produce a policy with a very large alerting volume and subsequently an operationally unacceptable alert rate. If two aircraft are relatively far away from each other and are moving away from each other, an alert is not prudent in most cases. Three states are considered to meet this information vector requirement: $\dot{\Delta h}, r_h, \alpha$.

To facilitate discussion, only R3 and CL1500/750 will be plotted henceforth; displaying all potential plots with the expanded state space easily becomes unwieldy. This is justified since the previous section discussed the general risk volumes for the different actions whereas R3 and L3 are similar and CL1500/750 and DES1500/750 are partial inverses.

Vertical range rate

Figure 5-24 and 5-25 shows the ν_{45} distributions for CL1500 and R3 for a $\{r_h, \Delta h, \dot{\Delta h}\}$ space. Foremost, the risk distribution along the Δh is dependent upon $\dot{\Delta h}$. If $\dot{\Delta h}$ is positive, then negative Δh states have significantly more risk since that vertical separation is decreasing. A similar trend is observed for a negative $\dot{\Delta h}$ and positive Δh . The distributions also indicate that regardless of the sign of Δh if $|\Delta h| \geq 1250$, the risk of an NMAC within 45 s is very low.

Risk is quantified vertically, with co-altitude $\Delta h = 0$ as a distinct inflection point between significant risk and little to no risk. This is especially evident when $\dot{\Delta h} = [25, 50]$. A long tail along r_h is evident since there is no information available quantifying the rate at which the risk changes along r_h . While seemingly common sense, this provides an analytic measure of NMAC risk when formulating the MDP.

How the two aircraft interact together, quantified by $\dot{\Delta}h$, is a better indicator of risk than what ownship is currently doing. While ownship climbing at a standard vertical rate above the intruder is generally very safe for manned aircraft, if $\dot{\Delta}h = [-50, -25]$ there is nontrivial risk. This highlights the importance and contribution of aircraft encounter system perspective

Figure 5.23 demonstrates a consequence of the UAS action set: more potential states can eventually transition to an NMAC. For small to moderate r_h values, the reduced climb rate leads to a smaller risk volume when $\dot{\Delta}h$ is positive. When $\dot{\Delta}h = [25, 50]$, the intruder must be moving away vertically since ownship can only produce a maximum $\dot{\Delta}h$ of 12.5 ft/s by itself. Conversely for $\dot{\Delta}h = [-50, -25]$, the intruder must be moving towards ownship vertically. These intruder assumptions are not valid for manned action set since $\dot{\Delta}h_o \in \pm 25$ ft/s.

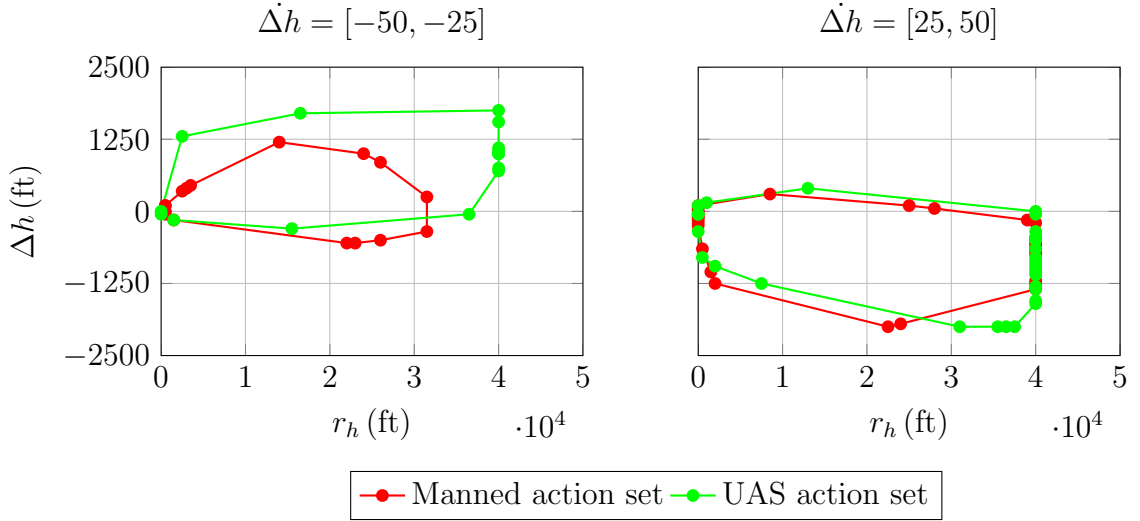


Figure 5.23: Comparison of $\nu_{30} \{r_h, \Delta h, \dot{\Delta}h \mid \text{CL1500/750}\}$

This signifies how the implicit information of a given state, such as $\dot{\Delta}h$ changes as a function of the action set. While the states and discretizations between the manned and UAS actions sets are identical, the risk associated with each state can vary and available assumptions about the aircraft encounter also change.

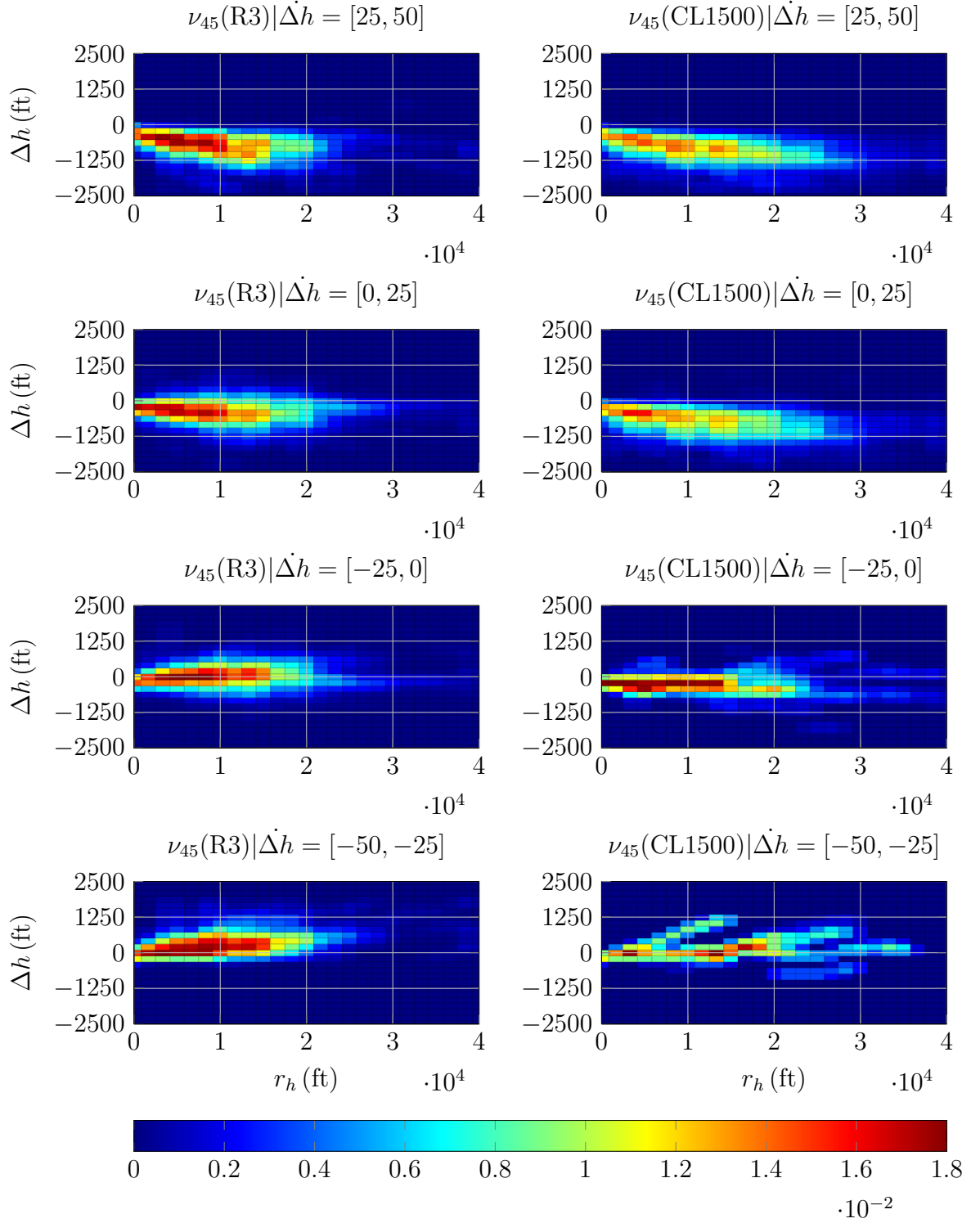


Figure 5.24: ν_{45} of manned action set for $\{r_h, \Delta h, \dot{\Delta h}\}$

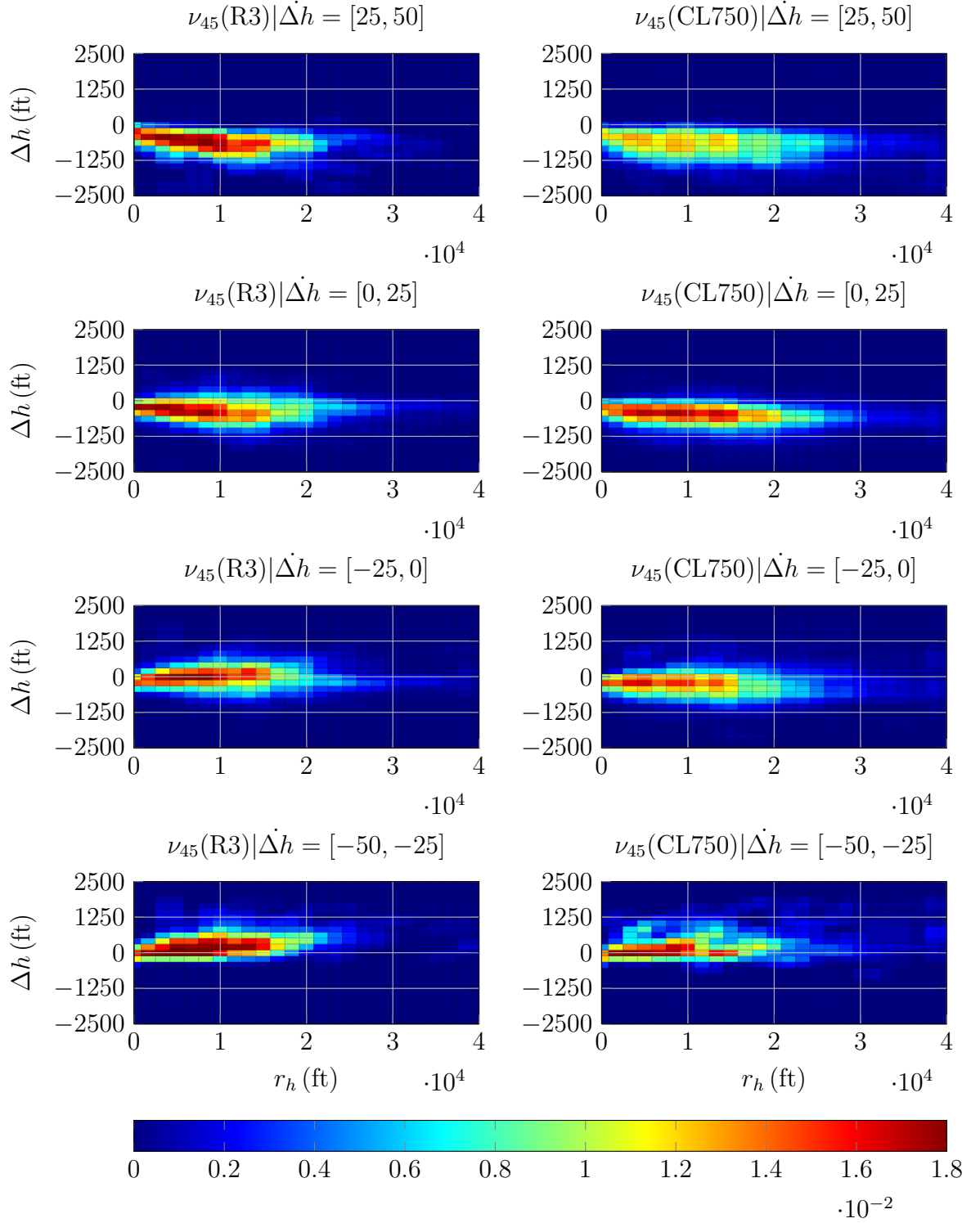


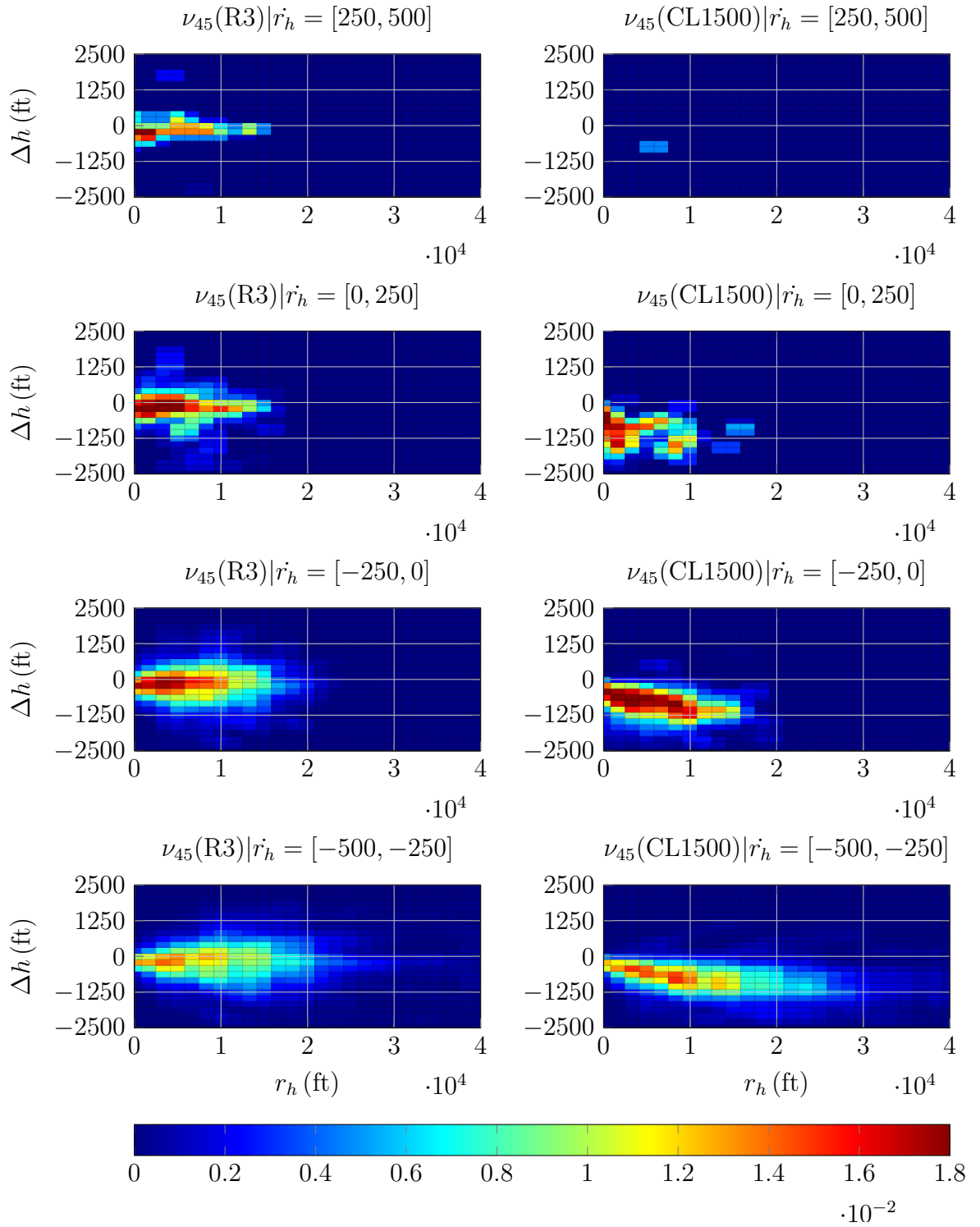
Figure 5.25: ν_{45} of UAS action set for $\{r_h, \Delta h, \dot{\Delta h}\}$

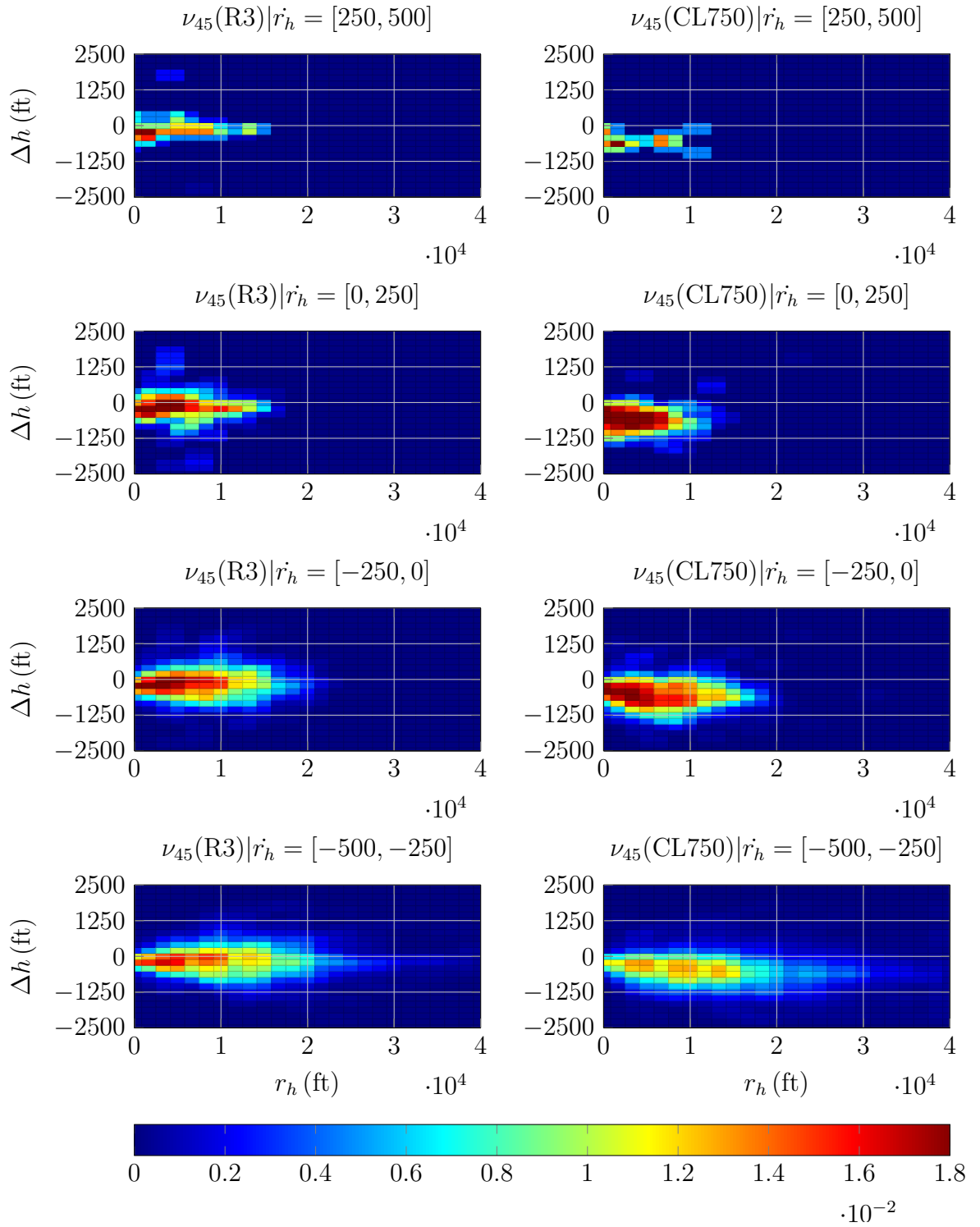
Horizontal range rate

Figure 5.26 and 5.27 shows the ν_{45} distributions for CL1500 and R3 when \dot{r}_h is added to the state space. A positive \dot{r}_h indicates that r_h is increase and the aircraft are moving away from each other. As \dot{r}_h becomes negative and there is greater risk at larger r_h state, the distribution becomes more similar to the baseline $\{r_h, \Delta h\}$ in Figures 5.18–5.19. As \dot{r}_h becomes more positive, the ν tail becomes smaller, resulting in a potentially smaller alerting region and alert rate. While, the $\{r_h, \Delta h\}$ effectively has to assume the worst case \dot{r}_h , inclusion of \dot{r}_h drastically changes the risk volumes and subsequently alerting characteristics in DP optimization.

The additional information provided by \dot{r}_h is very evident. If the ownship is climbing or descending (not shown) using the manned action set, there is almost zero risk of an NMAC when $\dot{r}_h = [250, 500]$, however risk remains if the aircraft is turning. This is due to the possibility of the ownship turning into the intruder and transiting to a negative \dot{r}_h .

While there is effectively nominal NMAC risk for $\dot{r}_h = [250, 500]$ with a manned vertical action, there is still noteworthy risk when using the UAS action set. There is also a more defined risk structure with the UAS action set when $\dot{r}_h = [0, 250]$, as evident by simply more NMAC events observed in those states. While the risk volume maybe greater than the manned, the inclusion of \dot{r}_h still drastically changes the risk volumes.

Figure 5.26: ν_{45} of manned action set for $\{r_h, \Delta h, \dot{r}_h\}$

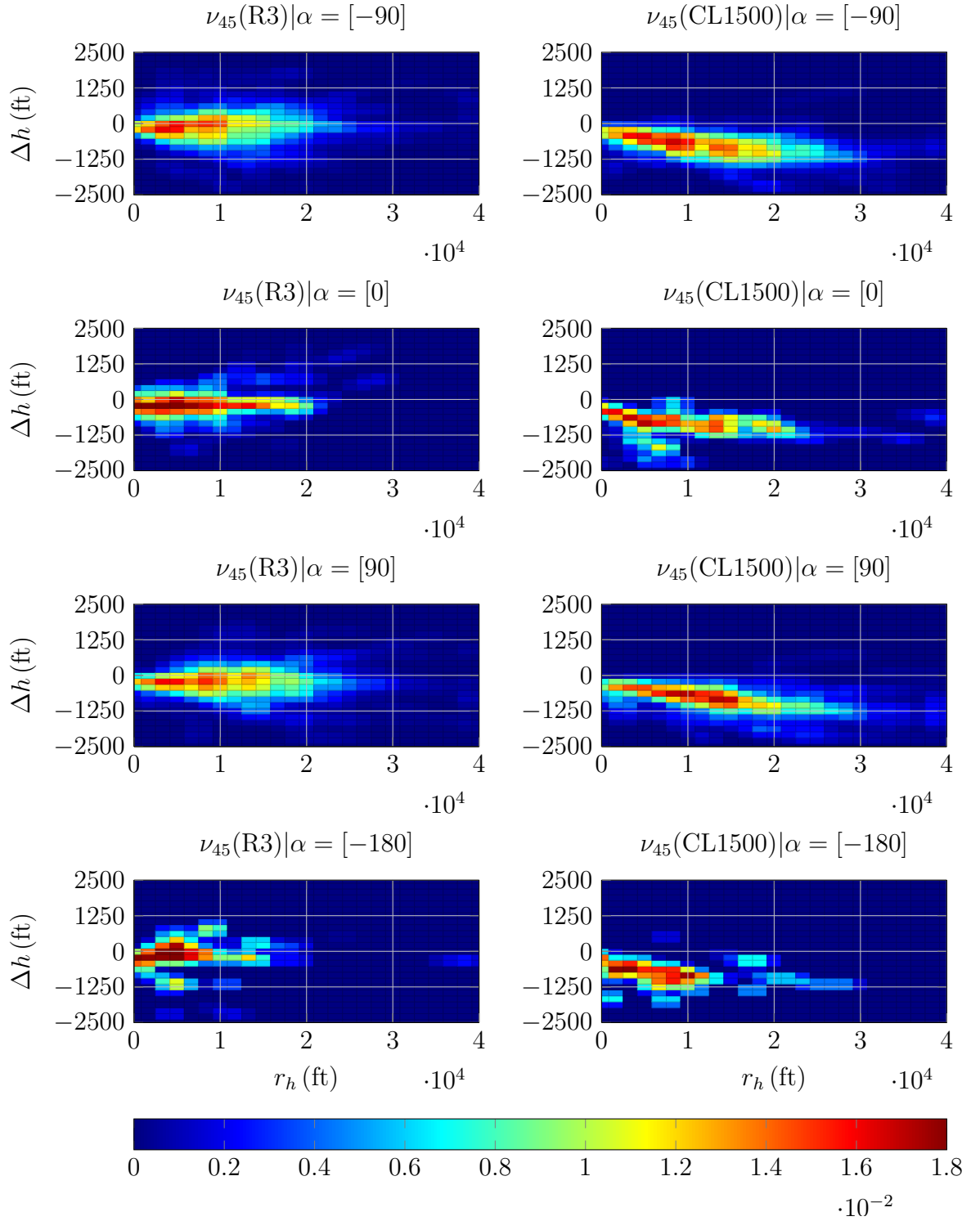
Figure 5.27: ν_{45} of UAS action set for $\{r_h, \Delta h, \dot{r}_h, \}$

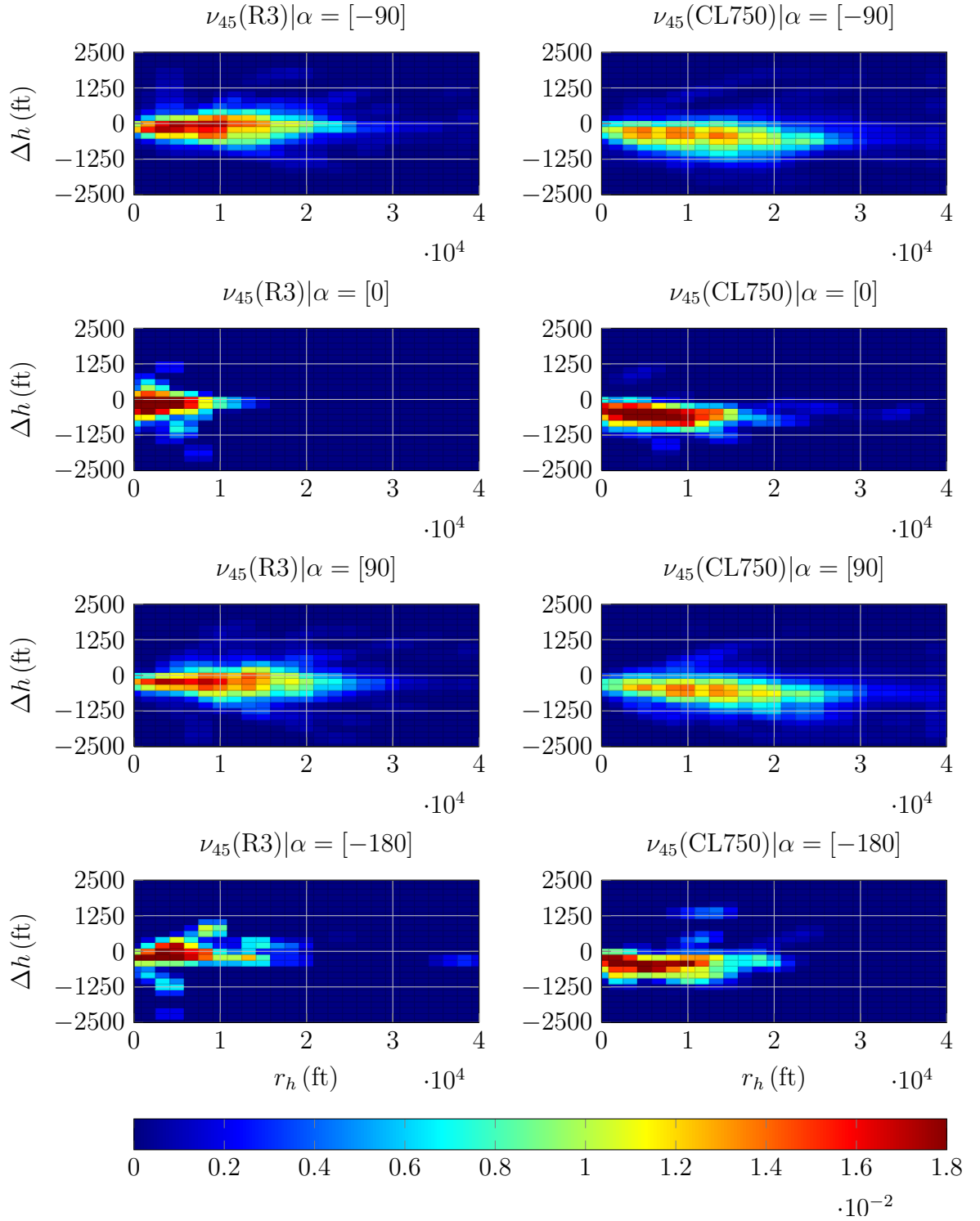
Bearing

Figure 5.29 and 5.29 show the ν_{45} distributions for CL1500 and R3 for a $\{r_h, \Delta h, \alpha, \}$ space. Risk given α is more subtle than the previous figures put still provides important information. When compared to r_h and Δh , α had the least differences between the two action sets. Since α encodes angular information, any action set uses the full α state range and no intruder assumptions can be made for specific α values or ranges. This is a consequence of the lack of sparsity of angular states, as discussed in Section 5.2.4.

Foremost, risk is similar if the intruder is directly left ($\alpha = 90$) or directly right ($\alpha = -90$) for both action sets and all maneuvers. Since α only contains angular information, it demonstrates the long r_h tail. The r_h tail is longer and with greater overall risk when using the UAS action set and when using vertical actions, regardless of the action set. Vertical actions, such as CL1500/750, have little relationship with α since \dot{h} operates independently to it a different axis. The long r_h and a less skewed distribution for CL1500/750, illustrates this relative independent between α and the vertical axis.

There is a distinct change in the distribution when the intruder is either head-on or overtaking ownship. For these geometries through the ownships nose, α provides significantly more information and the long r_h tails disappear. For vertical actions, the ν distributions becomes smaller with each nonzero element associated with a relatively greater probability of transitioning to a future NMAC state, as shown by the dense red clustering in the figures. However, as for $\alpha = \{-90, 90\}$, there is little difference between the action sets.

Figure 5.28: ν_{45} of manned action set for $\{r_h, \Delta h, \alpha, \}$

Figure 5-29: ν_{45} of UAS action set for $\{r_h, \Delta h, \alpha, \}$

5.4 Policy Generation

The previous sections analyzed the risk of NMAC through entropy and a time horizon for different state spaces. Using these metrics an MDP is formulated. Based on the identified risk, an alert region is calculated as convex hull and the DP algorithm is applied to the MDP. A baseline policy using the state space of $\{r_h, \Delta h\}$ is first generated to identify general trends about the cost and alerting structure. States for addition to the space are then identified using information theory. Evaluation for safety and operational feasibility of these policies is discussed in detail in Section 5.5.

5.4.1 Baseline

Using a policy-based DP optimization of a simple state space of $\{r_h, \Delta h\}$ with an equal alerting cost between horizontal and vertical maneuvers with the manned action set produces the collision avoidance policy in Figure 5-30. The ν_{30} was used to define the alerting region. Using the UAS alerting set produces a very similar policy. The policy relies heavily on vertical maneuvers and the lack of rate information results in the policy alerting out to 28,000 ft. This represents a very conservative and safe policy but is not operationally feasible, as latter discussed in Section 5.5.1.

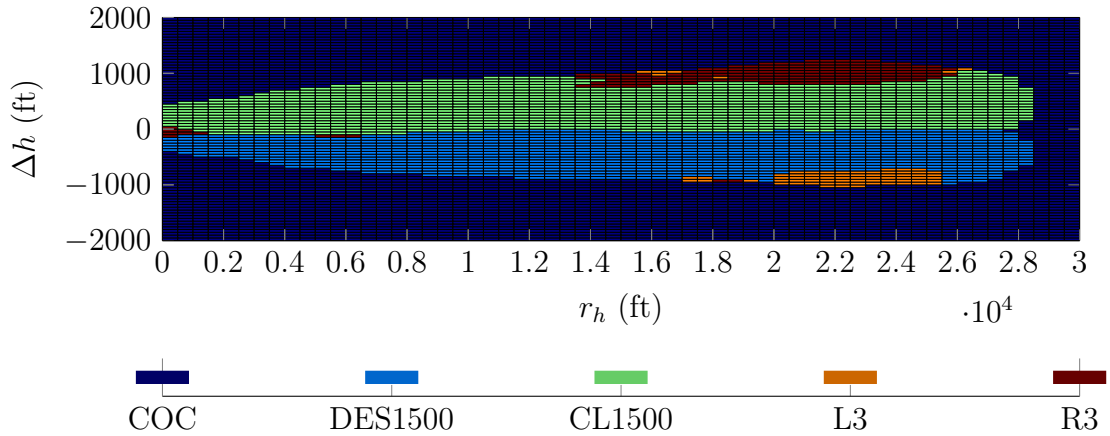


Figure 5-30: $\{r_h, \Delta h\}$ policy using a ν_{30} -based alerting

The horizontal alert cost (C_{ρ_h}) was decreased to explore how this influences the optimal policy. Figure 5.31 illustrates how the optimal policy of $\{r_h, \Delta h\}$, using the manned action set, changes as a function of the horizontal action alerting cost (C_{ρ_h}) as some smaller fraction of the vertical action alerting cost (C_{ρ_v}).

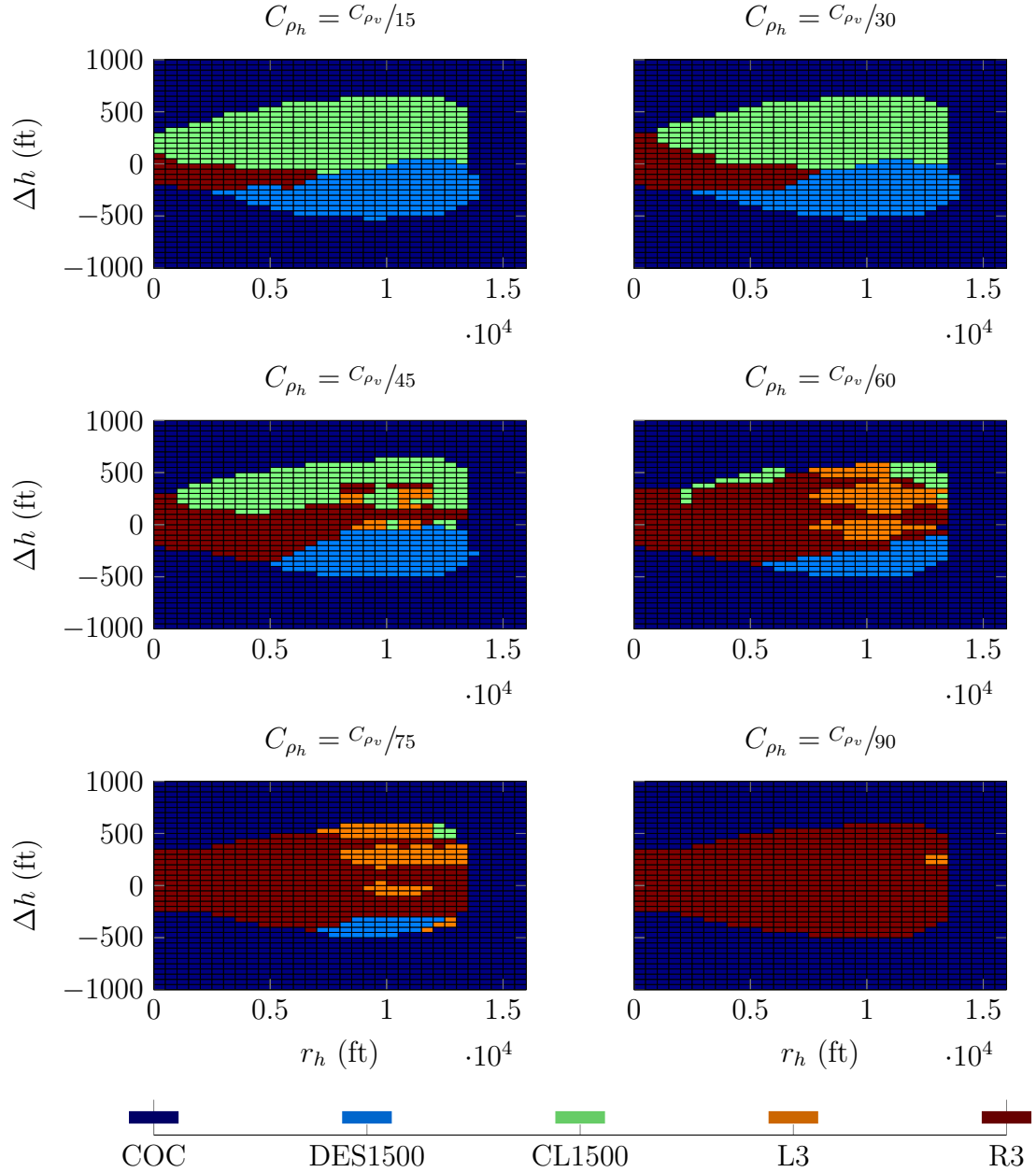


Figure 5.31: $\{r_h, \Delta h\}$ policies of varying C_{ρ_h} using a ν_{30} -based alerting

As C_{ρ_h} becomes smaller, there is a greater incentive to select a horizontal action. Horizontal actions “take over” the policy starting near NMAC and radiates outward along r_h . This isn’t surprising due to the serve cost penalty for NMAC states. Additionally, descend actions were replaced with a horizontal action faster than climb actions as C_{ρ_h} becomes smaller. This provides insight into to the cost-to-go values for each state and action; the values between DES1500/750 and R3 are closer than CL1500/750 and R3. This can be interpreted that there is stronger confidence that a climb will resolve the NMAC risk. Finally, without angular information, at the extreme case of $C_{\rho_h} = C_{\rho_v}/90$, only R3 is selected, outside a small noisy area of the policy. This lack of angular information is contributes to the irregular alert regions of L3, such as in $C_{\rho_h} = C_{\rho_v}/60$.

Although operational inefficient, this policy highlights many features found in more operationally acceptable policies. Foremost, a vertical maneuver of ± 25 ft/s is sufficient to maintain safety. Vertical maneuvers alone are sufficient as a collision avoidance action set; horizontal maneuvers require additional incentive to be optimally selected. This highlights why TCAS has been successful and ACAS Xa did not have to redesign the action space. More importantly, since the $\{r_h, \Delta h\}$ policy is conservative enough to meet safety feasibility, then the goal of adding additional states is to reduce the alert rate while secondarily increasing safety in specific cases (i.e. closely-space parallel approaches).

5.4.2 Adding States using Information Theory

While Section 5.3.2 discussed how adding states influenced ν_t and NMAC risk, ν_t alone doesn’t describe which state is more important. Information theory is used to determine what states to add to the baseline $\{r_h, \Delta h\}$ space. The NMAC entropy can range many bits depending upon the discretization. Thus when adding to the state space, both a state’s purpose and discretization need to be considered.

Various new three dimensional state spaces were generated by adding a single new state to the baseline $\{r_h, \Delta h\}$. These new state spaces were then compared to the baseline in terms of information gain and memory increase. An idea new space will minimize memory increase will maximizing information gain. Figure 5-32 compares these metrics for various spaces, with states classified as either horizontal or vertical.

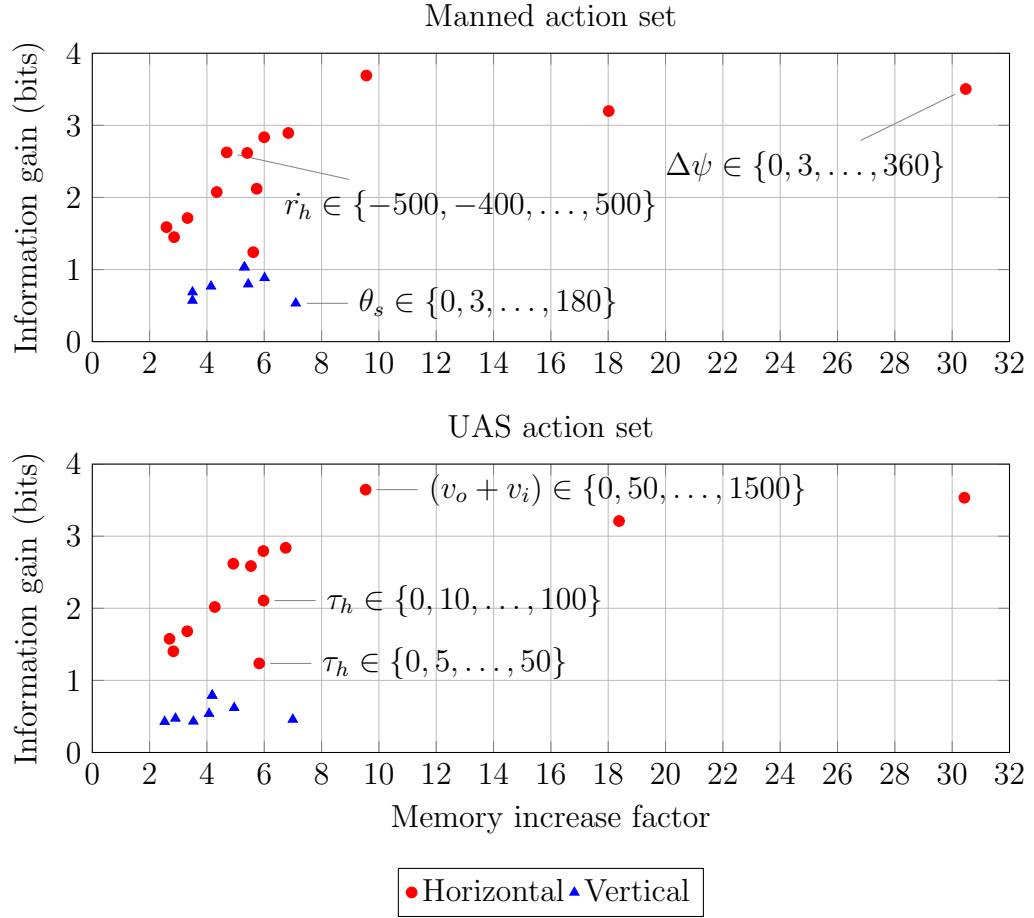


Figure 5-32: Gain and memory increase of adding a state to $\{r_h, \Delta h\}$

The general clustering and trends of the states are the same between action sets. Many of the conclusions from Section 5.2 continue to hold. Aggregate features such as $(v_o + v_i)$ or r_h generally provide more gain. It is also evident from a memory perspective, states relating to the horizontal plane generally yield greater gain.

States that independently yielded low NMAC entropy, such as θ_s , provided little

information gain. This supports the intuition that first-order states such as v or \dot{r}_h are more useful than θ_s when selecting states by hand. Additionally, compensating for low NMAC entropy with increased discretization can result in useful information gain but at the expense of an unacceptable memory increase. For example, $\Delta\psi \in \{0, 3, \dots, 360\}$ provides 3.5 bits of gain yet increases the memory requirements by a factor of 31. Yet, $\Delta\psi \in \{0, 6, \dots, 360\}$ provides a similar gain of 3.2 bits but only increased the memory requirements by a factor of 18.

Information gain can also potentially be increased by expanding the state's range with a coarser discretization. Both $\tau_{h,50} \in \{0, 5, \dots, 50\}$ and $\tau_{h,100} \in \{0, 10, \dots, 100\}$ have the same number of elements and have similar memory requirements, but the larger range results in greater gain. This highlights the complications associated with state maximums and saturation discussed in Section 3.5.3. Sometimes it is better to quantify a wide range at a low fidelity than a small range with a higher fidelity. The edge case of $\tau_h = 50$ encompasses many more encounter geometries than $\tau_h = 100$. This leads to greater saturation and provides less information as a result.

Based on the states tested, \dot{r}_h is recommended as the third state when using the either action set. \dot{r}_h shares mutual information with r_h and directly correlates with NMAC risk, as shown by ν_t . Airspeed states generally provided similar or slightly higher information gain but at the expense of a greater increase in memory. This isn't surprising because the horizontal plane requires more information to properly quantify it. Hence the the larger state space of ACAS Xu compared to ACAS Xa.

Figure 5-33 depicts parts of resulting policy of $\{r_h, \Delta h, \dot{r}_h\}$ where the alert costs are equal. Similar to $\{r_h, \Delta h\}$ there is a strong preference but vertical maneuvers. There is a stark difference in the alerting volume between $\dot{r}_h = -200$ and $\dot{r}_h = 100$, If the aircraft are moving away from each, the alerting region is significantly smaller.

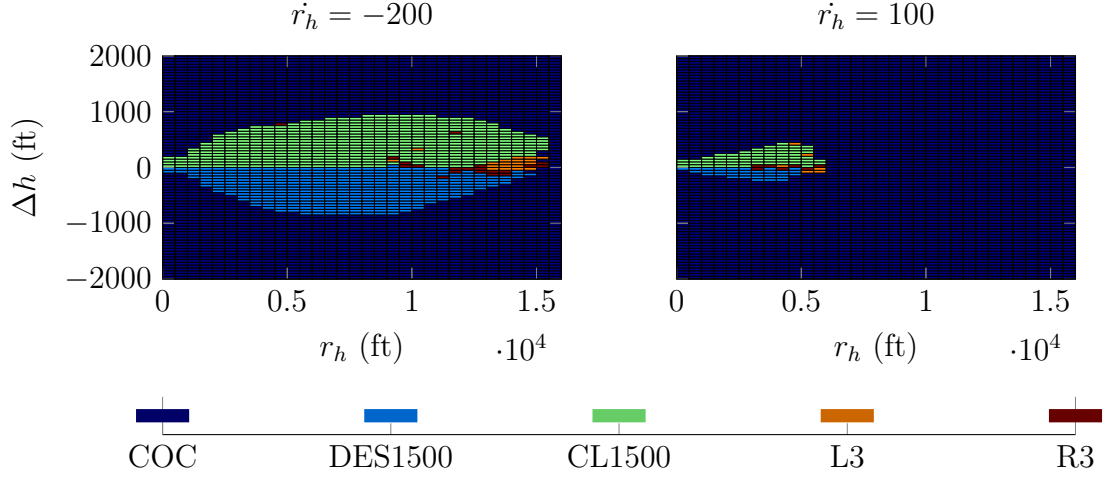


Figure 5.33: $\{r_h, \Delta h, \dot{r}_h \mid C_{\rho_h} = C_{\rho_v}\}$ policies using ν_{30} alerting

Unlike $\{r_h, \Delta h\}$ which met the safety but not operational requirements, this simple $\{r_h, \Delta h, \dot{r}_h\}$ policy yields an NMAC risk ratio of 0.1025 with an average alert duration of just 33 s. Section 5.5.2 discusses this evaluation in detail. The performance metrics will vary depending upon the alert costs, however the cost structure of $\{r_h, \Delta h\}$ doesn't directly correlate with $\{r_h, \Delta h, \dot{r}_h\}$ (or any other new state space).

5.4.3 Higher order states spaces

The process to add additional states is similar the approach described in Section 5.4.2 but this analysis did not automate the process. As Figure 5.23 illustrated, inclusion of $\dot{\Delta h}$ to the state space influences how the NMAC risk is perceived. Yet $\dot{\Delta h}$ has little NMAC entropy and produces significantly less information gain compared to other horizontal states. Vertical states have minimal mutual information with horizontal states, this presents a design challenge where high mutual information across the state space is desirable, leading to a minimization of conditional entropy.

Asserting that a complete space requires angular information and a vertical rate component, an angular state was selected as the fourth state variable because it provided greater independent entropy and information gain. For the fifth state, only

vertical states were considered. A discrete state variable is then added to account for COC transitions and strengthens and reversals.

$$S = \left\{ r_h, \Delta h, \dot{r}_h, \alpha, \left(\dot{h}_o + \dot{h}_i \right), s_{RA} \right\} \quad (5.5)$$

This space quantifies both the horizontal and vertical spaces as vectors with magnitude and angular components. Since the vertical axis is a line, a vertical angular state is not required. This policy doesn't quantify the ownship's current rate (i.e. \dot{h}_i) directly. With the joint action space, the aircraft have significantly more flexibility in responding to NMAC risk. Since ownship can maneuver either horizontally or vertically, ownship's current rates are less important.

For coordination, it is important that each aircraft have relatively unique perspectives or a “copycat” encounter occurs. If the state information for both aircraft are identical, then avoidance logics will issue identical maneuvers and the encounter will not resolve itself. The state's discretizations are almost as important as the states themselves. Consider an extreme case, $\Delta h = \{-8000, 8000\}$ provides little NMAC entropy and can not uniquely identify a co-altitude encounter geometry.

While furthering discretizing any state leads to diminishing returns, this is especially applicable to angular states due to the exponential growth of memory as the discretization becomes finer, illustrated by Figure 5-34 and observed in Section 5.2.4. While NMAC entropy can quantify the utility of different discretizations, it can't predict the optimal policy given a discretization due to costs and optimization parameters. It is important than to understand how the optimal policy changes with the discretization. For example, does a states optimal action change from depending upon the discretization? If the angular discretization is too course, are vertical maneuvers preferred because of the lack of information about the horizontal plane?

As the angular discretization becomes more course, the policy becomes more re-

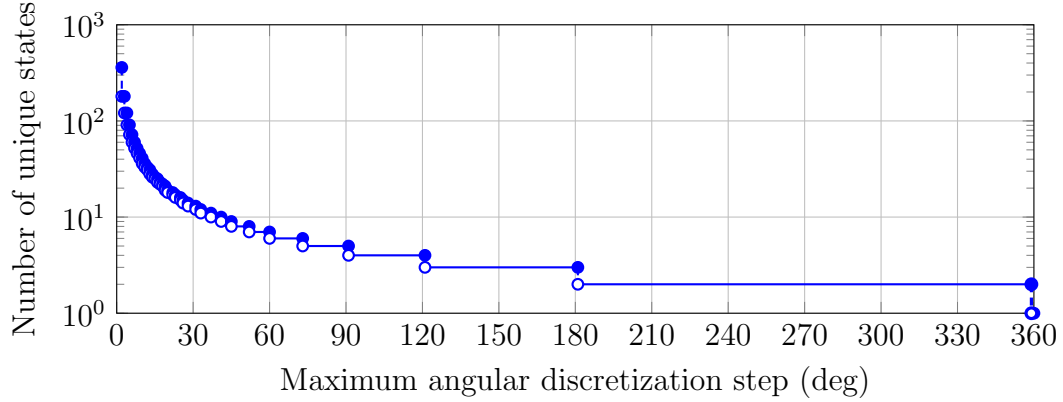


Figure 5-34: Angular discretization vs. unique states

liant on vertical maneuvers, similar to the optimal policies of $\{r_h, \Delta h\}$ and $\{r_h, \Delta h, r_h\}$. However, there is negligible difference between policies when the discretization is sufficiently small at a discretization of six degrees or less. This is advantageous because the memory requirements decrease as the discretization becomes more coarse. This is enabled by using simulation-based dynamics, in traditional MDP DP, it is possible can get “stuck” in a state if the action isn’t sufficient enough.

This is especially important for any angular or angular rate state such as α . The MDP must represent all four Euclidean quadrants, thus a discretized angular state must include $\{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$. This isn’t sufficient for horizontal maneuvers or for encoding right-of-way rules because a left or right sense is required, prompting the necessity of each 45° angle for a minimum of eight elements of any angular state. However, if only vertical maneuvers are implemented, like TCAS or ACAS Xa, angular information isn’t as relevant but not useless. Supplementing a τ state with basic angular information could improve a vertical action set MDP since additional information about the encounter geometry would be available.

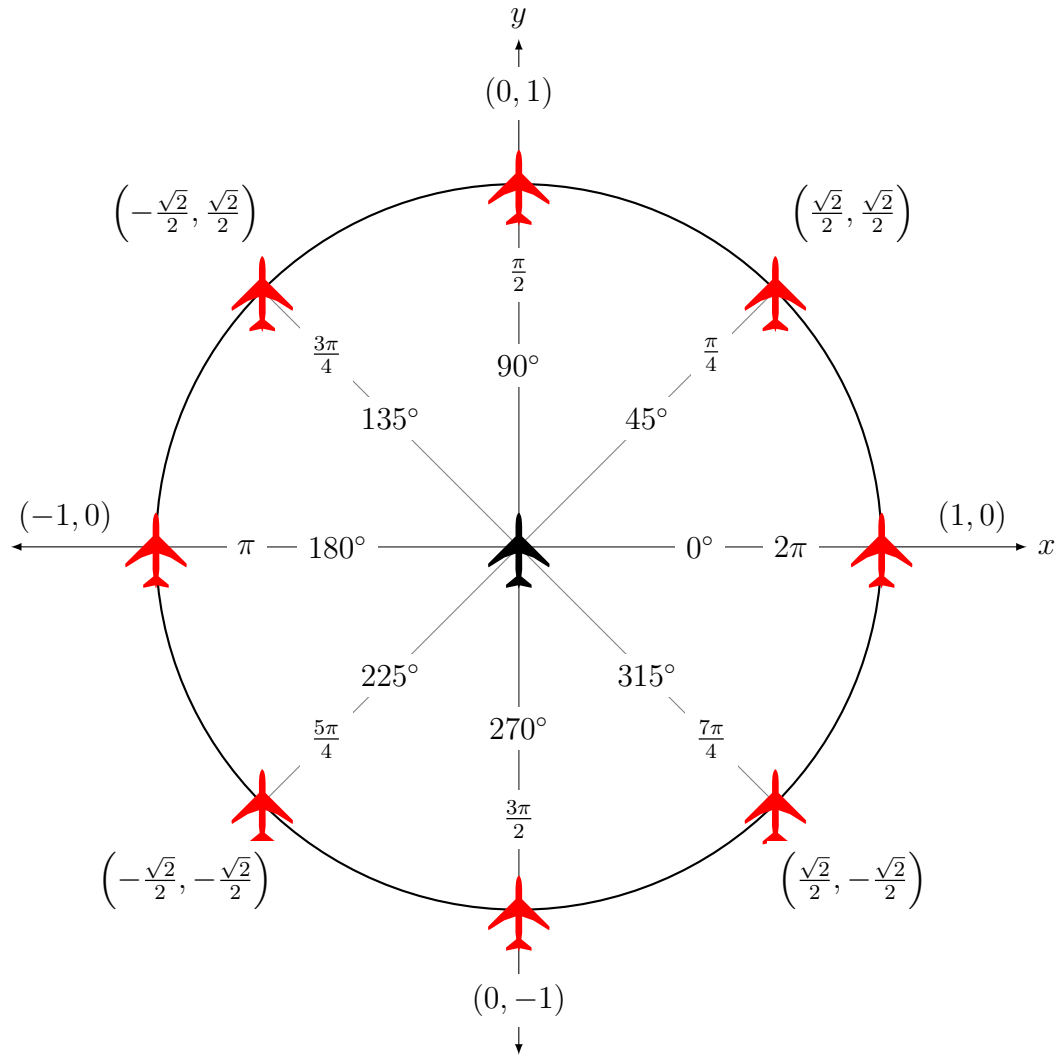


Figure 5-35: Required angular positions that the MDP must represent

5.4.4 Other Considerations

Section 3.2.2 defined the operational component of aircraft avoidance primarily as a function of alerting behavior, but there are many other operational components. Some of these components are passenger comfort, other aircraft types, and intended mission.

Performance Constraints

While horizontal maneuvers have anecdotally been associated with UASs, there is nothing prohibiting horizontal maneuvers for specific classes of manned aircraft. Previous sections have demonstrated that horizontal maneuvers can contribute to safe optimal policy, but these policies do not quantify any human conditions. Section 3.4 recognized that even for relatively slow airspeeds, the required bank angle for a $3^\circ/\text{sec}$ is greater than 30° which could affect passenger comfort or simply not be feasible.

Since the vertical actions differ between the sets, the operational performance constraints for different manned aircraft are ignored. Although a small four-seat general aviation has different operational performance constraints than a large cargo aircraft, this is not captured by the action sets. The ACAS X program addresses this by having four different variants, including one for general aviation. As discussed in Section 5.3, simply reducing the maximum vertical rate between the representative manned aircraft and UAS influenced NMAC risk. It is expected that changing the horizontal action would have similar effect and needs to be explored.

Rotorcraft and VTOL

States, such as \dot{h}_o or v_o , were not included in any evaluated policies but still could provide value. These states are particularly important when some of the assumptions are relaxed or removed, such as the fixed-wing aircraft assumption. To maintain controlled flight, a fixed-wing aircraft must have some positive airspeed. Rotorcraft, however, can operate in a hover mode where its airspeed is zero or fly “backwards” where the heading vector angle and airspeed vector angle are opposite.

This is important for aircraft avoidance because, for a given an RA, the resulting trajectory is dependent upon the aircraft’s airspeed vector. The resulting encounter geometry will evolve differently if the aircraft ascends or descends completely vertical

from a hover or when in motion. Additionally, the unique operating VTOL operating envelope allows for more potential controls to part of the action set. For example, a “stop and hover” action where the RA commands $v_o = 0$ ft/s and $\dot{h}_o = 0$ ft/s is feasible.

While rotorcraft possesses a unique operating envelope due to the ability to hover, during most phases of flight, the helicopter profile is similar to that of a general aviation airplane and rotorcraft pilots have expressed reluctance with vertical avoidance maneuvers (Taylor and Adams, 1985). When assessing TCAS for helicopters, it was asserted that the unique vertical dynamics of a helicopter do not significantly contribute to NMAC risk (Taylor and Adams, 1985). It would then be advantageous to have a similar collision avoidance MDP that is applicable for both general aviation and rotorcraft. If the MDPs were identification, the policies would not be memory optimal because there would be unreachable states in practice. For example, the rotorcraft MDP would need to quantify hover and somehow represent $v_o = 0$ ft/s, yet $v_o = 0$ ft/s isn’t applicable to a fixed-wing general aviation aircraft. This region of the state space would be incredibly more valuable for rotorcraft.

Intended Mission

By its very nature collision avoidance disrupts an aircraft’s intended mission. Commercial airlines want to achieve a certain heading to fly between distances, a vertical maneuver has minimal disruption on this intended mission. However, Navy aircraft such as the manned P-8A Poseidon or the Triton MQ-4C UAS have mission profiles include a dip down to a few thousand feet from a higher altitude (Steele, 2014) where a vertical maneuver can severely disrupt the intended mission. While JOCA addresses this by considering waypoints, a MDP DP approach requires expanding the state space or adding costs to quantify mission intent information.

The evaluated MDPs uses relative states ,such as Δh , to minimize memory requirements. While (h_o, h_i) can provide similar information as Δh , the range of (h_o, h_i)

is significantly larger. Whereas the range of Δh is thousands of feet, the range of h_o must be tens of thousands of feet since the MDP would represent all phases of flight: from take-off to a high altitude cruise. This wider range makes (h_o, h_i) is likely infeasible for MDP, regardless of how it correlates to a mission.

Adding a discrete mission state to the MDP would have a considerable effect on the memory required. Since this state would be discrete, sparsity can not be leveraged. The memory requirements would linearly increase as a function of the size of the mission state. Even a binary mission state (i.e. vertical transit and cruise) would double the memory requirements.

The simplest option is to develop multiple MDP and optimal DP policies for different missions and flight phases. This is an extension of the sensitivity options of TCAS and the procedure mode of ACAS X (Smith, 2013). The DP costs will change to promote SAA behavior that minimizes the disruption of the mission. For example, if the Triton MQ-4C UAS is executing a descend as part of a mission dip, the cost for climb can be relatively high or the horizontal maneuvers costs could be less. This will encourage behavior to maintain the descending profile while reducing NMAC risk.

5.5 Evaluation Results

Generated policies were evaluated in CASSATT and determined if safety and operationally feasible. Discussion focuses on the manned action set, with the UAS action set producing similar results. The safety metrics characterize how a policy mitigates NMACs and the operational assessment characterizes the alerting behavior. An ideal policy prevents all NMACs and has a low alert rate with alerts that last up to 30–60 s.

The baseline policy of $\{r_h, \Delta h\}$ was first evaluated, followed by $\{r_h, \Delta h, \dot{r}_h\}$ to highlight the benefits of adding just one state. Finally, $\left\{r_h, \Delta h, \dot{r}_h, \alpha, \left(\dot{h}_o + \dot{h}_i\right), s_{RA}\right\}$ is evaluated.

5.5.1 Baseline $\{r_h, \Delta h\}$

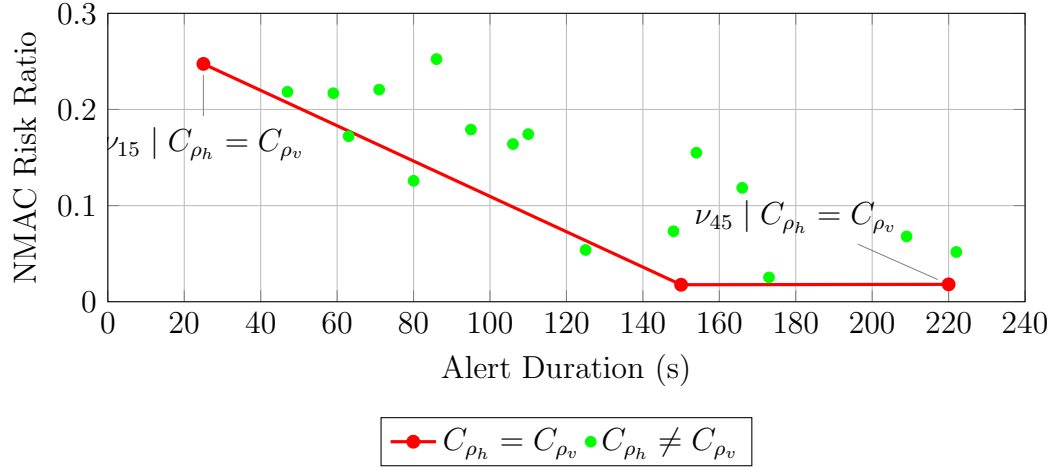
The conservative logic with equal horizontal and vertical maneuver costs with a state space consisting of $\{r_h, \Delta h\}$ discussed in Section 5.4.1, was first generated using various ν_t to define the alerting region. For the evaluation, only the ownship is equipped with a SAA system and observes the intruder via simulated simple white noise tracker. Table 5.2 reports the Monte Carlo evaluation results.

Table 5.2: Baseline $\{r_h, \Delta h\}$ results with manned action space

ν_t horizon	NMAC Risk Ratio	5NMAC Risk Ratio	Mean Alert Duration (s)
15	0.2474	0.3851	25
30	0.0177	0.1607	150
45	0.0181	0.0665	220

As mentioned in Section 5.4, this baseline meets the safety requirements but not the operational requirements. However relying on last-second maneuvers, as dictated by the smaller alerting region of ν_{15} , reduces the NMAC risk by approximately 75% and protects against the larger 5NMAC volume by approximately 60%. This indicates that many of the “easy” NMAC encounters can be resolved with minimal state information and warning before NMAC. Doubling the risk projection from 15 s to 30 s drastically reduces the safety risk but at the expense at a higher alert rate.

The relationship between safety and alert is illustrated by Figure 5-36, which plots the $C_{\rho_h} = C_{\rho_v}$ results from Table 5.2 and selection of other policies generated where $C_{\rho_h} \neq C_{\rho_v}$ and with different filtering of ν_t . Since the baseline policy of $\{r_h, \Delta h \mid C_{\rho_h} = C_{\rho_v}\}$ from Section 5.4.1, relies on almost solely on vertical maneuvers, few MDPs where $C_{\rho_h} > C_{\rho_v}$ were considered. For $\{r_h, \Delta h\}$, vertical maneuvers do not have to be incentivized against horizontal maneuvers.

Figure 5-36: Results using $\{r_h, \Delta h\}$

Self separation implications

The resolution of at least 60% of 5NMAC events with a simple policy is interesting from an UAS SAA perspective, which consists of collision avoidance and self separation. While the 5NMAC volume is smaller than most proposed well-clear volumes that self separation protects, it does provide insight into state requirements for self separation. The policy is designed to minimize NMAC risk, yet the policy significantly reduces the risk of aircraft transitioning close to an NMAC. This suggestion that defining risk based on a smaller volume can potentially produce an optimal policy that meets the safety requirements for a large volume. It is expected that the alert duration and alert rate will be smaller than if risk was defined by the desired large volume. This hypothesis applies for any aircraft avoidance MDP with costs defined by risk.

If feasible, this can lead to a new methodology to defining self separation MDPs and development of the self separation functionality. Unlike collision avoidance which must resolve an overwhelmingly majority of potential NMACs, the maximum failure rate of self separation is less stringent. The less rigorous safety requirement may only require a smaller fidelity MDP representation of the aircraft encounter.

It is conceivable that self separation functionality can be added to a collision avoidance policy through additional costs. This functionality may only use a subset of the collision avoidance states to meet computational and memory requirements. This concept contrasts the current independent MDPs and implementation of collision avoidance and self separation. It would be useful to evaluate states based on self separation entropy and a well-clear risk-based horizon. It is possible that utility of a state changes based on the SAA functionality.

Consequence of only using two states

The lack of angular or rate information significantly impacts the alerting characteristics regardless of the action space. Specifically, for encounters where $\dot{h}_o \sim \dot{h}_i$ and $\Delta \dot{h} = \epsilon$, the ownship can't transition out of the alerting region vertically and will continue to alert until the horizontal alerting region is cleared. This can be from a few to many nautical miles, leading to the long alert durations. It is especially dangerous if the aircraft are also parallel, since the baseline policy relies primarily on vertical maneuvers, it won't transition out of the parallel situation. Similarly for overtaking and head-on encounters, if $\Delta \dot{h} = \epsilon$, the policy will continue to alert prior to and after the aircraft pass each other at CPA.

The lack of information on how the encounter is evolving is the other major contributor to the long alert duration. Whether or not the aircraft are moving towards or away from each other has little influence on the policy. As Figure 5-33 from Section 5.4.2 illustrated, ν_t changes with r_h and the corresponding alerting region can drastically shrink. Inclusion of rate information should then have a significant impact on the alert duration but without angular information will still strongly prefer vertical maneuvers when $C_{\rho_h} = C_{\rho_v}$.

5.5.2 Effect of adding range rate

Section 5.4.2 described the processing of adding a third state to create the $\{r_h, \Delta h, \dot{r}_h\}$ space. Table 5.3 provides the evaluation metrics when all the alert costs are equal. Using just three states, both the safety and operational feasible metrics can be met with the following uniform state space that requires only 1.4637 Mb of memory:

$$\Delta h = \{-2000, -1950, \dots, 2000\} \quad (5.6)$$

$$r_h = \{0, 500, \dots, 20000\} \quad (5.7)$$

$$\dot{r}_h = \{-500, -400, \dots, 500\} \quad (5.8)$$

This demonstrates that defining the alert volume, an operational component, as a function of ν_t , a safety component, is feasible when designing a collision avoidance advisory logic. This primarily due to that for relatively positive \dot{r}_h that the NMAC risk is relatively low which significantly shrinks the alert volume, leading to reduction in alerting. The course discretization is supported by the joint-action space, ownship is not only reliant on horizontal maneuvers to resolve an encounter. Instead of finely discretizing either the horizontal or vertical, a moderate to course discretization is sufficient for both.

Table 5.3: $\{r_h, \Delta h, \dot{r}_h \mid C_{\rho_h} = C_{\rho_v}\}$ results with manned action space

ν_t horizon	NMAC Risk Ratio	5NMAC Risk Ratio	Mean Alert Duration (s)
15	0.1843	0.2134	14
30	0.1025	0.1356	33
45	0.0562	0.0733	50

Note that $\{r_h, \Delta h, \dot{r}_h\}$ meets the objectives without an angular component because when $C_{\rho_h} \cong C_{\rho_v}$, the horizontal maneuvers that require an angular component are rarely deemed optimal.

Effect of \dot{r}_h on alerting behavior

Inclusion of \dot{r}_h resulted in substantially shorter alert durations without compromising meeting the safety objectives. For ν_{30} and ν_{45} , the mean duration decreased by at least 400%. In particular, slow overtaking and head-on encounters had much shorter alert durations. The slow closure rates provide the aircraft more time to execute another nominal maneuver (i.e. something on the flight plan) before the NMAC risk becomes sufficient enough to warrant an alert.

Additionally, slow or positive \dot{r}_h require less adjustment to the encounter geometry to minimize NMAC risk. For example, Figure 5-33 from Section 5.4.2 shows that the vertical component of the policy is much smaller when $\dot{r}_h = 100$ than when $\dot{r}_h < 100$. If the policy relies primarily on vertical maneuvers and the desired change of Δh is relatively small, then it simply will take less time to achieve the desired Δh than if the desired Δh was large.

The policy can also exhibit some oblique behavior due to the alerting regions defined by NMAC risk usually when the alert does not resolve the NMAC risk. Consider a head-on encounter when $\dot{r}_h \cong -500$ and the optimal policy issues an RA. However if $\dot{r}_h \cong 500$, the optimal policy may never issue an RA. Suppose during this head-on encounter, that regardless of the RA the aircraft cross paths. Once they cross, \dot{r}_h will jump from -500 ft/s to 500 ft/s and stop instantaneously stop alerting. This behavior also affects any τ state where τ can jump from a small positive number to its edge state or ∞ and angular states where the state jumps between quadrants. Ironically, this behavior does not occur with the simpler $\{r_h, \Delta h\}$ space because of coupled dynamics of the states; there are no quadrants or asymptotic events to “jump” over.

Effect of not including angular information

While the alerting characteristics of overtaking and head-on encounters changed, the lack of angular information however makes it impossible to distinguish between these encounter types with parallel encounters. This lack of information also results in a policy primarily composed of vertical maneuvers and still produces some conservative behavior. Specifically, crossing encounters cannot be distinguished between overtaking or head-on encounters. Depending upon the geometry, during a crossing encounter the intruder aircraft can “nick” the front edge of the alerting region resulting in a few second alert, which would probably be considered a nuisance to manned pilots.

Therefore, meeting the feasibility metrics doesn’t indicate that a $\{r_h, \Delta h, \dot{r}_h\}$ is sufficient for operational use. The inability to unique represent common encounter types can be addressed through adding additional states. Rather it demonstrates the utility of a system perspective, the benefits of not decomposing into τ , and the flexibility of the joint-action space.

5.5.3 Full state space

Section 5.4.3 outlined the methodology of creating a complete state representation of $\left\{r_h, \Delta h, \dot{r}_h, \alpha, \left(\dot{h}_o + \dot{h}_i\right), s_{RA}\right\}$. The complexity of separate vertical and horizontal alert costs presented a difficult and new challenge compared to ACAS X development, where alerts generally have one cost. This space was built upon the moderately discretized space of $\{r_h, \Delta h, \dot{r}_h \mid C_{\rho_h} = C_{\rho_v}\}$. It added an angular component with bearing and a vertical rate component with the aggregate summation of vertical rates:

$$\Delta h = \{-2000, -1950, \dots, 2000\} \quad (5.9)$$

$$r_h = \{0, 500, \dots, 20000\} \quad (5.10)$$

$$\dot{r}_h = \{-500, -400, \dots, 500\} \quad (5.11)$$

$$\alpha = \{-180, -135, \dots, 180\} \quad (5.12)$$

$$(\dot{h}_o + \dot{h}_i) = \{-50, 25, \dots, 50\} \quad (5.13)$$

Like the smaller state spaces, this state achieved the safety and operational feasibility metrics. However, this policy did not as strongly favor vertical maneuvers, due to inclusion of angular information. Specifically, horizontal maneuvers significantly influenced α which led to better cost-to-go values.

Table 5.4: $\left\{ r_h, \Delta h, \dot{r}_h, \alpha, (\dot{h}_o + \dot{h}_i), s_{RA} \mid C_{\rho_h} = C_{\rho_v} \right\}$ results

ν_t horizon	NMAC Risk Ratio	5NMAC Risk Ratio	Mean Alert Duration (s)
15	0.1758	0.2031	20
30	0.0834	0.1256	36
45	0.0651	0.0937	55

Inclusion of the two additional states increased the memory requirements from 1.4637 Mb to 41.4043 Mb. Thus, adding states such as \dot{h}_o are not memory prohibitive. However the simulation-based framework becomes more susceptible to memory constraints as the state space grows. Storing the state-transition matrices of spaces with tens of millions unique combinations can be prohibitive, even with sparse representations. For example, this space has 1,035,045 unique state combinations. The state-transition matrix is then size $1,035,045 \times 1,035,045$ with 1,071,318,152,025 elements stored as doubles. While memory savings can be realized by using a float data representation, the large-scale of state-transition matrices can be problematic for the simulation-based approach.

This policy can be viewed as a refinement of $\{r_h, \Delta h, r_h\}$ in that many of the general conclusions and discussion points are similar between the policies but that $\left\{r_h, \Delta h, r_h, \alpha, \left(\dot{h}_o + \dot{h}_i\right), s_{RA}\right\}$ can simply better distinguish between encounter geometries. Including angular information is a key driver for and it is expected that a different angular state than α would also achieve this. It is emphasized however that addition of these states is to meet operational feasibility alone and that there no safety reason to prefer horizontal maneuvers or vertical.

The simulation-based framework was never intended to replace the traditional MDP DP approach of ACAS X, but rather provide contributions as an algorithmic design framework. This framework has shown that defining risk as a function of NMACs, state selection via information theory, and the acceptance that not every state requires a fine discretization leads to feasible and reasonable developmental SAA policies.

Chapter 6

Conclusion

Collision avoidance is a critical aviation safety technology and has been under development since the 1950's. The evolving ATM system has led to new user classes, new surveillance technologies, and new operating modes. With the introduction of UASs into the airspace, many historical assumptions are no longer valid. Hence, the existing collision avoidance solution, TCAS, is no longer sufficient. This is partly due to TCAS's pseudocode architecture, making it difficult to adapt the technology to current needs. In response, ACAS X is being developed as TCAS's successor and leverages MDPs to quantify the aircraft avoidance problem and DP optimization to generate a robust and adaptive advisory logic.

Due to the computational requirements for DP, it has been difficult to produce an advisory logic that jointly considers horizontal and vertical maneuvers. Vertical and horizontal maneuvers require different information vectors for optimization, a simple union of these information vectors isn't feasible due to the "curse of dimensionality." Additionally, there has yet been an extensive and complete assessment of potential aircraft avoidance MDP states with regards to computational requirements and relationship to a goal state, such as an NMAC. Furthermore majority of previous collision avoidance research has leveraged vertical maneuvers, leading to less of an overall understanding on how horizontal maneuvers influence an aircraft encounter.

In response, a simulation-based framework was developed to better understand how each potential state quantifies the aircraft avoidance problem with regards to

safety and operational components. Using information theoretics, computational signals processing, and statistics, the NMAC entropy and NMAC horizon metrics were developed and calculated. The analysis leveraged, to the author’s knowledge, the single largest aircraft avoidance simulation data set ever generated of approximately 6,970,080,000 numeric doubles. Using these metrics, an collision avoidance MDPs was formulated and evaluated for feasibility.

The primary research objectives of analytically quantifying the individual components of an aircraft avoidance MDP and the first demonstration of a joint horizontal and vertical action set MDP were met due to developing of a simulation-based framework and new applications of information theory to collision avoidance. Future work will emphasize developing of collision avoidance for other airframes, such as VTOLs, and development of more advanced information theoretic metrics.

6.1 Contributions of Simulation-Framework

ACAS X has been a very successful research and development effort that is paving the way forward for the next generation of operational aircraft avoidance. The initial clear focus towards manned aircraft, led to some design decisions to limit the flexibility of the developed framework. Specifically, the barrier of the curse of dimensionality and the challenges to developing a horizontal and vertical action policy. The contributed simulation-based approach leverages the concept of MDP DP while introducing new flexibility for algorithmic design.

6.1.1 Memory management

A key assumption to the simulation-based approach is that the state dynamics, regardless of the dimensionality or composition, will be loosely-coupled with a high degree of sparsity. If true, this will allow the state-transition and cost matrices to be represented in a sparse format with the realization of substantial memory savings.

Unrealistic or unreachable states will be zero value elements in the sparse matrix. They did not have to be considered during optimization.

The ability to calculate the state-transitions is dependent upon the ability to produce enough state samples to build a distributions. Only recently through D4M and high performance computing is this possible. Storage alone was not sufficient, the processing code had to be computationally efficient and fast. Sparse matrices and pre-calculated state-transition and cost matrices enabled this fast processing and optimization. In terms of memory management this approach was very direct and simple, a key distinction in making the machine DP process “human readable.”

Since sparse matrices only store the indices of non-zero elements, there is no penalty for allocating an extremely large state space with a large span. This enabled the true maximum and minimum potential states to be identified and ability to fully construct the tails of the distribution. During state exploration, the risk of saturating the edge states was nonexistent. Based on this analysis, an appropriate state span and discretization for a MDP was identified.

6.1.2 State and action selection

As previous noted in Section 3.4, aircraft are increasingly being controlled in different ways. Since the simulation-model captures the dynamics, generating the state-transitions is a matter of calculating conditional probabilities. Dynamic propagation was not required. This enabled the rapid development of different state spaces and action sets.

Two representative action sets were generated and analyzed in parallel using the same code base. A direct comparison between performance capabilities with regards to NMAC risk was completed. A new analytical relationship between an aircraft’s performance and the logic’s alerting behavior was identified and explored. This analysis was only recently feasible due to advances in parallel signals processing.

This approach can easily be extended to VTOLs and rotorcraft. Whereas ACAS X is currently focused on fixed-wing aircraft, there is no variant for rotorcraft. The dynamic capability to hover and strafe need to be account for and there doesn't exist a rotorcraft centric aircraft encounter model.

6.1.3 Delay and RA state

The ACAS X framework considers response delay during the optimization as delay is not directly captured by the encounter models. For the described approach, if the Monte Carlo simulation includes delay, this delay will be captured in the state-transitions. Since delay is captured by the state-transitions, it does not necessary require its own state. This directly affects the s_{RA} variable found in TCAS and ACAS X which must account for pilot delay. With the simulation-based approach, s_{RA} reduces to the state itself and no longer needs to “countdown” till the desired action is executed.

A drawback of this approach is that a new simulation is required to implement a probabilistic response model. Different response models can not be added dynamically. Changes to the simulation must be accounted for at the start of the simulation-based processes, whereas ACAS X implement these changes when optimizing.

6.2 Contributions of Information Theoretics

Information theory and the concept of entropy had sporadically been applied to aviation safety and was mostly applied for strategic problems. There was only a few instances of applying these concepts to a tactical aviation problem like collision avoidance. The development and successful use of the NMAC entropy and NMAC horizon metrics demonstrated the viability and potential of applying information theoretics to aviation safety problems. Specific contributions were made in quantifying risk and assessing an MDP prior to optimization and policy assessment.

6.2.1 Quantifying risk of NMAC

Previous research used a simulation-based approach to establish a risk-based separation standard for UAS (Weibel et al., 2011) but only used considered a small number of states and didn't consider all joint probabilities between states. The developed framework demonstrated the ability to calculate conditional probabilities for state and conditional probabilities for any combination of states. This contribution lead to the development of the NMAC horizon variable, ν_t that quantifies the probability of an NMAC within t seconds.

The NMAC entropy of a state provided a one-step perspective of the state's utility. Calculating and analyzing the NMAC horizon (ν_t) provided a longer multi-step perspective with regards to safety. Figure 5-18 expands beyond this previous research by identifying NMAC risk as a function of position and ownship's action. Together, they demonstrated that defining the alert volume, an operational component, as a function of ν_t , a safety component, is feasible when designing a collision avoidance advisory logic.

6.2.2 Surrogate for MDP state's utility

NMAC entropy and information gain was demonstrated to be surrogate for the potential utility of a state as part of a MDP formulation. States such as θ_s provided little NMAC entropy and little mathematical or operational justification for inclusion into the MDP. Historically recognized important states such as Δh and r_h had some the largest NMAC entropy values.

Beyond just identifying states, the information theoretic metrics enabled a human-readable assessment of different discretizations. Insights into questions such as, "Which state Is it better to finely discretize?" can now be analytically and quickly obtained without optimizing and evaluating an MDP.

6.3 Future Work

The optimal policies described in Section 5.4 demonstrate the feasibility of the proposed state spaces but are not a replacement for any ACAS X variant at this time. These policies were evaluated only against one type of aircraft encounters and have not been stressed against other important types, such as closely spaced parallel approaches. Instead of repeating many of the safety evaluations of ACAS X, the described research can be leveraged by ACAS X to address online costs and memory requirements. Specifically, ν_t should be explored as an online cost and the applying the concept of sparsity to could reduce memory requirements.

The different performance characteristics of VTOL compared to a fixed-wing aircraft can potential lead to a different aircraft avoidance MDP formulation. Different assumptions will result in different state ranges and discretizations; specifically VTOLs need to account for zero airspeed while fixed-wing aircraft do not. The NMAC entropy and NMAC horizon analyses were intentionally generic with minimal fixed-wing aircraft assumptions to facilitate the early development of collision avoidance (CA) system for VTOLs. As previous mentioned, the simulation-based framework can assist in rapid and flexible development of new algorithms, since there is little historical precedent or design to build an algorithm from. Instead of modifying the optimization framework, with the simulation-based approach only helicopter dynamics need to be developed. The lack of a specific encounter model is mitigated through Monte Carlo simulations that could best approximate encounter situations and have the helicopter dynamics provide realistic transition propagation. This requires substantially less code development and provides a more accurate solution than manually developing an encounter model through expert opinion.

The developed NMAC entropy and NMAC horizon metrics can also be applied further to aviation safety assessments for any system. Understanding how TCAS,

ACAS X, or JOCA change the probability of an NMAC given some time horizon is incredibly useful. It provides a more detailed assessment of the system than relying on summary statistics or encounter type assessments. The newly development metrics can be correlated or integrated with historically accepted metrics to add another level of scrutiny to these human-safety systems.

Finally, determining the utility of any state is challenging for many problems other than aviation collision avoidance. With the advent of Big Data, it has become very easy to record and store large data sets with many states but processing them remains a challenge. This is particularly a concern for Public Safety systems and communication networks, which in disaster and incident scenarios can be just as safety critical as an aircraft avoidance system. Applying information theoretic concepts to Public Safety decision points or scenarios has significant potential and should be explored.

Appendix A

Simulation States

This appendix reports the states collected from the Monte Carlo simulations. Any of these states can be considered as part of potential MDP, along with any aggregated features developed from these states. Each state is recorded as a floating point number with position-based states at 1.0 precision; rate magnitudes (i.e. vertical rate, airspeed) at 1.0 precision; angular rates at 0.001 precision; and accelerations at 0.001 precision. The following states were recorded for all aircraft:

- Airspeed
- Airspeed acceleration
- Altitude
- Altitude rate of change
- Bank angle
- Body-fixed angular vector (p, q, r)
- Commanded vertical rate
- Commanded turn rate
- Commanded longitude acceleration
- East position
- East position rate
- Heading angle
- North position
- North position rate
- Pitch angle
- Pitch rate
- Turn rate
- Vertical acceleration
- Vertical rate
- Yaw angle
- Yaw rate

Additionally, the following encounter states were recorded:

- Encounter id
- Encounter weight
- Horizontal range
- NMAC
- Simulation id
- Time
- Vertical range

Appendix B

Encounter Geometry Types

This chapter illustrates many common types of encounter geometries between two aircraft. These geometries are frequently referenced in Chapter 5 when discussing if a specific state can help uniquely identify a specific geometry. Only geometries decomposed into the vertical axis or horizontal plane are discussed. For example, a complex encounter where an aircraft spirals into another is not specifically classified. Lastly, only geometries that can lead to a loss of separation are presented.

B.1 Vertical

Figure B.1 illustrates the three basic geometries that can potentially lead to an NMAC. A co-altitude encounter occurs when all vertical rates are zero. Additionally, one aircraft can either climb or descend into another level aircraft or both aircraft can climb or descend into each other.

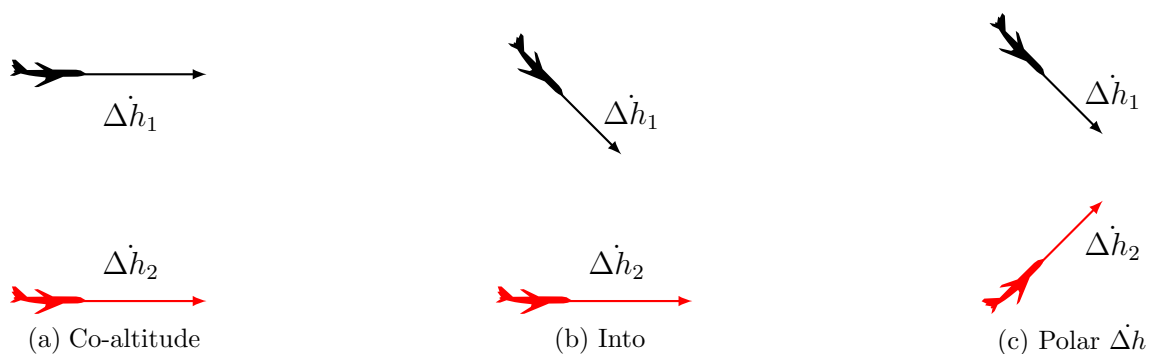


Figure B.1: Vertical encounters

B.2 Horizontal

Figure B.2 illustrates basic head-on and overtaking encounters. A head-on encounter occurs when $\psi_1 \sim -\psi_2$ regardless of their airspeeds. The overtaking encounter occurs when $\psi_1 \sim \psi_2$ and the airspeed of the aircraft “behind” is greater.



Figure B.2: Head-on and overtaking encounters

Figure B.3 illustrates crossing and parallel encounters, which are typically more challenging to resolve. The crossing encounter are moving towards the same airspeed but with perpendicular headings $\psi_1 \perp \psi_2$. Conversely, a parallel encounter occurs when the aircraft are side-by-side with $\psi_1 \sim \psi_2$ and $v_1 \sim v_2$. Closely-spaced parallel encounters when, in general, $r_h \leq 3000$ ft are generally considered one of the more dangerous types of encounters.



Figure B.3: Crossing and parallel encounters

Finally, Figure B·4 illustrates a slow-closing encounter occurs where $0 < |\psi_1 - \psi_2| \leq \epsilon$. The two aircraft are nearly parallel but with a small relative heading between them. The encounter geometry will eventually produce an NMAC but at a relatively longer time-scale.

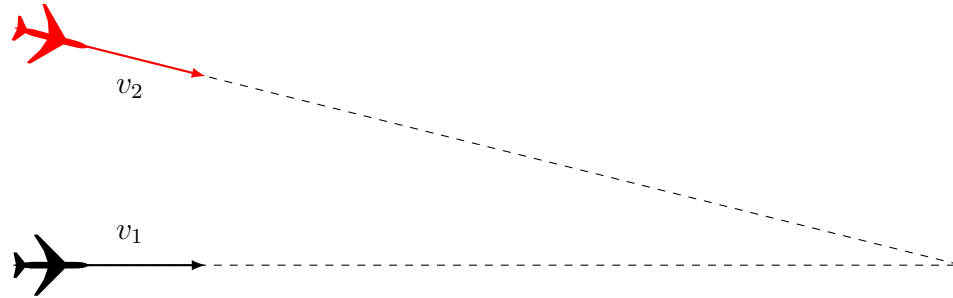


Figure B·4: Slow-closure encounter

References

- Apache Software Foundation (2014). Apache accumulo. <https://accumulo.apache.org/>.
- April, J., Glover, F., Kelly, J. P., and Laguna, M. (2003). Practical introduction to simulation optimization. In *Proceedings of the 2003 Winter Simulation Conference*, volume 1, pages 71–78. IEEE.
- Asmar, D. and Kochenderfer, M. (2013). Optimized airborne collision avoidance in mixed equipage environments. Project Report ATC-408, Massachusetts Institute of Technology, Lincoln Laboratory, Lexington, MA.
- Audet, C., Booker, A. J., Dennis Jr, J., Frank, P. D., and Moore, D. W. (2000). A surrogate-model-based method for constrained optimization. *8th AIAA / USAF / NASA / ISSMO Symposium on Multidisciplinary Analysis and Optimization*. doi: 10.2514/6.2000-4891.
- Bai, H., Hsu, D., Kochenderfer, M. J., and Lee, W. S. (2012). Unmanned aircraft collision avoidance using continuous-state pomdps. In *Robotics: Science and Systems VII*, pages 1–8.
- Bejan, A. (1995). *Entropy generation minimization: the method of thermodynamic optimization of finite-size systems and finite-time processes*. CRC press.
- Bellman, R. (1956). Dynamic programming and lagrange multipliers. *Proceedings of the National Academy of Sciences of the United States of America*, 42(10):767.
- Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517.
- Bertsekas, D. P. (2005). *Dynamic programming and optimal control*, volume 1-2. Athena Scientific Belmont, MA, 3 edition.
- Bertsekas, D. P. and Tsitsiklis, J. N. (1995). Neuro-dynamic programming: an overview. In *Proceedings of the 34th IEEE Conference on Decision and Control*, volume 1, pages 560–564. IEEE.
- Billingsley, T. B., Kochenderfer, M. J., and Chryssanthacopoulos, J. P. (2012). Collision avoidance for general aviation. *IEEE Aerospace and Electronic Systems Magazine*, 27(7):4–12.

- Black, P. E. (2014). big-o notation. <http://xlinux.nist.gov/dads//>.
- Bliss, N. T. and Kepner, J. (2007). 'pmatlab parallel matlab library'. *International Journal of High Performance Computing Applications*, 21(3):336–359.
- Box, G. E. and Wilson, K. (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 13(1):1–45.
- Brooker, P. (2013). Introducing unmanned aircraft systems into a high reliability atc system. *Journal of Navigation*, 66(05):719–735.
- Burgess, D. W. and Altman, S. I. (1995). Tcas iii bearing error evaluation. ATC Report ATC-231, MIT Lincoln Laboratory, Lexington, MA.
- Burgess, D. W., Altman, S. I., and Wood, M. L. (1994). Tcas: Maneuvering aircraft in the horizontal plane. *Lincoln Laboratory Journal*, 2:295–312.
- Byun, C., Arcand, W., Bestor, D., Bergeron, B., Hubbell, M., Kepner, J., McCabe, A., Michaleas, P., Mullen, J., O’Gwynn, D., Plount, A., Reuther, A., Rosa, A., and Yee, C. (2012). Driving big data with big compute. In *2012 IEEE Conference on High Performance Extreme Computing (HPEC)*, pages 1–6, Waltham, MA. IEEE.
- Cattell, R. (2011). Scalable SQL and NoSQL data stores. *ACM SIGMOD Record*, 39(4):1227.
- Chadés, I., Cros, M., Garcia, F., and Sabbadin, R. (2005). Markov decision process (mdp) toolbox v2. 0 for matlab. *INRA Toulouse, INRA, France*, <http://www.inra.fr/internet/Departements/MIA/T/MDPtoolbox>.
- Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A., and Gruber, R. E. (2008). Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems*, 26(2):1–26.
- Chen, R., Gevorkian, A., Fung, A., Chen, W.-Z., and Raska, V. (2011). Multi-sensor data integration for autonomous sense and avoid. In *Infotech@Aerospace*. American Institute of Aeronautics and Astronautics. doi:10.2514/6.2011-1479.
- Cheng, C.-H. (1997). Evaluating naval tactical missile systems by fuzzy ahp based on the grade value of membership function. *European Journal of Operational Research*, 96(2):343–350.
- Chryssanthacopoulos, J. P. and Kochenderfer, M. J. (2011). Collision avoidance system optimization with probabilistic pilot response models. In *American Control Conference*, pages 2765–2770, San Francisco, California.

- Chryssanthacopoulos, J. P. and Kochenderfer, M. J. (2012). Decomposition methods for optimized collision avoidance with multiple threats. *Journal of Guidance, Control, and Dynamics*, 35(2):398–405.
- Clark, J. and McFarland, A. (1977). Initial collision avoidance algorithms for the beacon-based collision avoidance system. Technical Report FAA-RD-77-163, The MITRE Corporation.
- Code of Federal Regulations (2011). Title 14, chapter 91, section 113,.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., et al. (2001). *Introduction to algorithms*, volume 2. MIT press Cambridge.
- Davies, S. (1997). Multidimensional triangulation and interpolation for reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1005–1011.
- de Farias, D. P. and Van Roy, B. (2006). A cost-shaping linear program for average-cost approximate dynamic programming with performance guarantees. *Mathematics of Operations Research*, 31(3):597–620.
- de Farias, D. P. and Weber, T. (2008). Choosing the cost vector of the linear programming approach to approximate dynamic programming. In *47th IEEE Conference on Decision and Control, 2008*, pages 67–72. IEEE.
- Department of Defense (2007). Use of international airspace by U.S. military aircraft and for missile/projectile firings. DoD Instruction 4540.01.
- DeWitt, D. and Gray, J. (1992). Parallel database systems: the future of high performance database systems. *Communications of the ACM*, 35(6):85–98.
- Doebbler, J., Gesting, P., and Valasek, J. (2005). Real-time path planning and terrain obstacle avoidance for general aviation aircraft. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*. doi:10.2514/6.2005-5825.
- Doye, J. P. and Wales, D. J. (1998). Thermodynamics of global optimization. *Physical review letters*, 80(7):1357.
- Edwards, M. (2012). A safety driven approach to the development of an airborne sense and avoid system. In *AIAA Infotech@Aerospace Conference*. doi 10.2514 / 6.2012-2485.
- Edwards, W. M. (2010). Encounter models for the littoral regions of the national airspace system. Technical Report CASSATT-2, Massachusetts Institute of Technology, Lincoln Laboratory.

- Espindle, L. P., Griffith, J. D., and Kuchar, J. K. (2009). Safety analysis of upgrading to tcas version 7.1 using the 2008 u.s. correlated encounter model. Technical report, Project Report ATC-349, Massachusetts Institute of Technology, Lincoln Laboratory.
- European Organisation for the Safety of Air Navigation (2011). Acas ii post-implementation safety case. Technical report, European Organisation for the Safety of Air Navigation.
- Fang, S.-C., Rajasekera, J. R., and Tsao, H.-S. J. (1997). *Entropy optimization and mathematical programming*. Springer.
- Federal Aviation Administration (2013). Unmanned aircraft systems (uas) operational approval. http://www.faa.gov/documentLibrary/media/Notice/N_8900.227.pdf.
- Federal Aviation Administration (2013). Sense and Avoid (SAA) for Unmanned Aircraft Systems (UAS): Second caucus workshop report.
- Fernández, J. L., Sanz, R., Simmons, R. G., and Diéguez, A. R. (2006). Heuristic anytime approaches to stochastic decision processes. *Journal of Heuristics*, 12(3):181–209.
- Gertz, J. (1983). Mode-S surveillance netting. Technical report, MIT-LIN-ATC-120, MIT Lincoln Laboratory.
- Graham, S., Chen, W.-Z., De Luca, J., Kay, J., Deschenes, M., Weingarten, N., Raska, V., and Lee, X. (2011). Multiple intruder autonomous avoidance flight test. In *AIAA Infotech@ Aerospace Technical Conference, St. Louis, MO*. doi:10.2514 / 6.2011-1420.
- Grappel, R. (2001). ASR-9 processor augmentation card (9-PAC) Phase II scan-scan correlator algorithms. Technical report, MIT-LIN-ATC-298, MIT Lincoln Laboratory.
- Griffith, J., Edwards, M. W., Miraflor, R. M., and Weinert, A. (2013). Due Regard Encounter Model Version 1.0. Project Report ATC-397, Massachusetts Institute of Technology, Lincoln Laboratory.
- Griffith, J. D. and Edwards, M. (2010). Evaluation of the Jointly Optimal Collision Avoidance Algorithm (JOCA) Version 2.0 with U.S. Airspace Encounter Models. Technical Report ATC-358, Massachusetts Institute of Technology, Lincoln Laboratory.
- Griffith, J. D., Kochenderfer, M. J., and Kuchar, J. K. (2008). Electro-optical system analysis for sense and avoid. In *AIAA Guidance, Navigation and Control Conference*. doi:10.2514/6.2008-7253.

- Griffith, J. D. and Olson, W. (2011). Coordinating general aviation maneuvers with TCAS Resolution Advisories. Technical report, Project Report ATC-374, Massachusetts Institute of Technology, Lincoln Laboratory.
- Harman, W. (1989). Tcas- a system for preventing midair collisions. *The Lincoln Laboratory Journal*, 2:437–458.
- Hauskrecht, M. (2000). Value-function approximations for partially observable markov decision processes. *Journal of Artificial Intelligence Research*, 13(1):33–94.
- Holland, J. E., Kochenderfer, M. J., and Olson, W. A. (2013). Optimizing the Next Generation Collision Avoidance System for Safe, Suitable, and Acceptable Operational Performance. In *Tenth USA/Europe Air Traffic Management Research and Development Seminar*.
- ICAO (1990). Rules of the air. In *Annex 2 to the Convention on International Civil Aviation*.
- Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492.
- Kepner, J. (2009). *Parallel MATLAB for multicore and multinode computers*, volume 21. SIAM.
- Kepner, J., Anderson, C., Arcand, W., Bestor, D., Bergeron, B., Byun, C., Hubbell, M., Michaleas, P., Mullen, J., and O’Gwynn, D. (2013). D4M 2.0 schema: A general purpose high performance schema for the accumulo database. In *IEEE High Performance Extreme Computing Conference (HPEC)*, page 16, Waltham, MA. IEEE.
- Kinnan, L. M. (2009). Use of multicore processors in avionics systems and its potential impact on implementation and certification. In *IEEE/AIAA 28th Digital Avionics Systems Conference*, pages 1–E. IEEE.
- Kochenderfer, M. J. and Chryssanthacopoulos, J. P. (2011). Robust airborne collision avoidance through dynamic programming. Project Report ATC-371, MIT Lincoln Laboratory.
- Kochenderfer, M. J., Chryssanthacopoulos, J. P., Kaelbling, L. P., and Lozano-Perez, T. (2010a). Model-based optimization of airborne collision avoidance logic. Project Report ATC-360, Massachusetts Institute of Technology, Lincoln Laboratory.
- Kochenderfer, M. J., Chryssanthacopoulos, J. P., and Weibel, R. E. (2012). A new approach for designing safer collision avoidance systems. *Air Traffic Control Quarterly*, 20(1):27.

- Kochenderfer, M. J., Edwards, M. W. M., Espindle, L. P., Kuchar, J. K., and Griffith, J. D. (2010b). Airspace encounter models for estimating collision risk. *Journal of Guidance, Control, and Dynamics*, 33(2):487–499.
- Kochenderfer, M. J., Espindle, L. P., Edwards, M., Kuchar, J. K., and Griffith, J. D. (2009). Airspace encounter models for conventional and unconventional aircraft. In *Eighth USA/Europe Air Traffic Management Research and Development Seminar (ATM2009)*.
- Kochenderfer, M. J., Espindle, L. P., Kuchar, J. K., and Griffith, J. D. (2008a). Correlated encounter model for cooperative aircraft in the national airspace system. Project Report ATC-344, Massachusetts Institute of Technology, Lincoln Laboratory.
- Kochenderfer, M. J., Griffith, J. D., and Olszta, J. (2010c). On estimating mid-air collision risk. In *the Tenth AIAA Aviation Technology, Integration, and Operations Conference*.
- Kochenderfer, M. J., Kuchar, J. K., Espindle, L. P., and Griffith, J. D. (2008b). Uncorrelated encounter model of the National Airspace System version 1.0. Project Report ATC-345, MIT Lincoln Laboratory, Lexington, Massachusetts.
- Kuchar, J. (2005). Safety analysis methodology for unmanned aerial vehicle (UAV) collision avoidance systems. In *USA/Europe Air Traffic Management R&D Seminars*, Baltimore, MD.
- Kuchar, J. E. and Drumm, A. C. (2007). The traffic alert and collision avoidance system. *Lincoln Laboratory Journal*, 16(2):277.
- Kullback, S. (1997). *Information theory and statistics*. Courier Dover Publications.
- Lacher, A. R., Maroney, D. R., and Zeitlin, A. D. (2007). Unmanned aircraft collision avoidance—technology assessment and evaluation methods. In *The 7th Air Traffic Management Research & Development Seminar Barcelona, Spain*.
- Laguna, M. and Martí, R. (2002). Neural network prediction in a system for optimizing simulations. *IIE Transactions*, 34(3):273–282.
- Lai, J. (2010). *A hidden Markov model and relative entropy rate approach to vision-based dim target detection for UAV sense-and-avoid*. PhD thesis, Queensland University of Technology.
- Law, A. M., Kelton, W. D., and Kelton, W. D. (1991). *Simulation modeling and analysis*, volume 2. McGraw-Hill New York.

- Littman, M. L., Cassandra, A. R., and Kaelbling, L. P. (1995). Learning policies for partially observable environments: Scaling up. In *International Conference on Machine Learning*, pages 362–370.
- Lutz, R., Hanrahan, T., Schneider, D., Edwards, M., Graeff, R., and Gould, N. (2013). Advanced modeling and simulation techniques for evaluating system-of-systems performance. In *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*, volume 2013. NTSA.
- Lv, J., Zhang, X., and Guan, X. (2013). A conflict avoidance approach based on memetic algorithm under 4d-trajectory operation concept. In *IEEE/AIAA 32nd Digital Avionics Systems Conference (DASC)*, pages 6A2–1. IEEE.
- McCarley, J. S. and Wickens, C. D. (2004). Human factors concerns in uav flight. <http://www.hf.faa.gov/hfportalnew/Search/DOCs/uavFY04Planrpt.pdf>.
- McLaughlin, M. P. (1997). Safety study of the Traffic Alert and Collision Avoidance System (TCAS II). Technical Report MTR 97W32, MITRE Corporation.
- Mehlhorn, K. and Sanders, P. (2008). *Algorithms and data structures: The basic toolbox*. Springer.
- Metropolis, N. and Ulam, S. (1949). The monte carlo method. *Journal of the American statistical association*, 44(247):335–341.
- MITRE, C. (1983). System Safety Study of Minimum TCAS II. Technical report, MTR-83W241, McLean, VA.
- Mon, D.-L., Cheng, C.-H., and Lin, J.-C. (1994). Evaluating weapon system using fuzzy analytic hierarchy process based on entropy weight. *Fuzzy sets and systems*, 62(2):127–134.
- Morrel, J. (1956). Fundamental physics of the aircraft collision problem. Technical Memo 465-1016-39, Bendix Corporation.
- Munos, R. and Moore, A. (2002). Variable resolution discretization in optimal control. *Machine learning*, 49(2-3):291–323.
- Murphy, K. P. (2002). *Dynamic bayesian networks: representation, inference and learning*. PhD thesis, University of California, Berkeley.
- Myers, R. H., Montgomery, D. C., and Anderson-Cook, C. M. (2009). *Response surface methodology: process and product optimization using designed experiments*. John Wiley & Sons, 3 edition.
- Organick, E. I. (1966). *A Fortran IV primer*. Addison-Wesley, Reading, MA.

- Papoulis, A. and Pillai, S. U. (2002). *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education.
- Pissanetzky, S. (1984). *Sparse matrix technology*. Academic Press.
- Powell, W. B. (2007). *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons.
- Pritchett, A. R., Fleming, E. S., Cleveland, W. P., Popescu, V. M., Thakkar, D. A., and Zoetrum, J. J. (2012). Pilots information use during tcas events, and relationship to compliance to tcas resolution advisories. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 56, pages 26–30. Sage Publications.
- Puterman, M. L. (2009). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.
- Rasmussen, J. (1983). Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man and Cybernetics*, pages 257–266.
- Ross, S., Pineau, J., Paquet, S., and Chaib-Draa, B. (2008). Online planning algorithms for pomdps. *Journal of Artificial Intelligence Research*, 32:663–704.
- Schlager, N. (1994). *When technology fails: Significant technological disasters, accidents, and failures of the twentieth century*. Gale Group.
- Shakernia, O., Chen, W., and Raska, V. (2005). Passive Ranging for UAV Sense and Avoid Applications. *Infotech@ Aerospace*. doi:10.2514/6.2005-7179.
- Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55.
- Sinsky, A., Reed, J., and Fee, J. (1984). Enhanced TCAS II tracking accuracy. In *AIAA 6th Digital Avionics Systems Conference*, volume 3.
- Šišlák, D., Jisl, P., Volf, P., Pechoucek, M., Nicholson, D., Woodhouse, D., and Suri, N. (2009). Integration of probability collectives for collision avoidance in agentfly. In *Proceedings of 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 69–76.
- Smith, K. A. (2013). *Collision avoidance system optimization for closely spaced parallel operations through surrogate modeling*. PhD thesis, Massachusetts Institute of Technology.
- Steele, J. (2014). Next chapter for navy drone. <http://www.utsandiego.com/news/2014/aug/20/Northrop-triton-navy-drone-palmdale/>.

- Stonebraker, M. (2010). SQL databases v. NoSQL databases. *Communications of the ACM*, 53(4):10.
- Taylor, F. and Adams, R. (1985). Helicopter User Survey – Traffic Alert and Collision Avoidance System (TCAS). Final Report DOT/FAA/PM-85/6, Systems Control Technology, Inc.
- Temizer, S., Kochenderfer, M., Kaelbling, L., Lozano-Perez, T., and Kuchar, J. (2009). Unmanned aircraft collision avoidance using partially observable markov decision processes (project report atc-356). Project Report ATC-356, Massachusetts Institute of Technology, Lincoln Laboratory, Lexington, MA.
- Torczon, V. (1997). On the convergence of pattern search algorithms. *SIAM Journal on Optimization*, 7(1):1–25.
- Weibel, R. E., Edwards, M. W. M., and Fernandes, C. S. (2011). Establishing a risk-based separation standard for unmanned aircraft self separation. In *Ninth USA/Europe Air Traffic Management Research & Development Seminar*.
- Weinert, A. J., Harkleroad, E. P., Griffith, J. D., Edwards, M. W., and Kochenderfer, M. J. (2013). Uncorrelated encounter model of the national Airspace system version 2.0. Project Report ATC-404, Massachusetts Institute of Technology, Lincoln Laboratory, Lexington, MA.
- William, H. (1989). TCAS: a system for preventing midair collisions. *Lincoln Laboratory Journal*, 2(3):437–458.
- Williams, K. W. (2007). An assessment of pilot control interfaces for unmanned aircraft. Final Report DOT/FAA/AM-07/8, Federal Aviation Administration.
- Williamson, T. and Spencer, N. A. (1989). Development and operation of the Traffic Alert and Collision Avoidance System (TCAS). *Proceedings of the IEEE*, 77(11):1735–1744.
- Wood, M. L. (1987). Choosing between Vertical and Horizontal Escapes. Unpublished Technical Memo.
- Yuster, R. and Zwick, U. (2005). Fast sparse matrix multiplication. *ACM Transactions on Algorithms (TALG)*, 1(1):2–13.
- Zeitlin, A. D. (1979). Active beacon collision avoidance system – collision avoidance algorithms. MITRE Technical Report MTR-79W00110, The MITRE Corporation, McLean, Virginia.

CURRICULUM VITAE

Andrew Joseph Weinert

Research Interests

- Artificial intelligence
- Computational science
- Public Safety information and communication systems
- Unmanned aircraft airspace integration

Education

Boston University Master's of Science, January 2015
College: Electrical and Computer Engineering
Concentration: Systems and Control

The Pennsylvania State University Bachelor's of Science, December 2009
Major: Security Risk and Analysis
Minors: IST for Aerospace Engineering, Natural Science
Certificate: Information Science and Technology

Graduate Coursework

EC 501: Dynamic Systems Theory

EC 505: Stochastic Processes

EC 521: Cybersecurity

EC 524: Optimization Theory and Methods

EC 700: Computation Neuroscience Seminar

EC 710: Dynamic Programming

EC 745: Nanomedicine: Principles and Applications

Research Affiliations

- MIT Lincoln Laboratory (MIT LL)

Professional Affiliations

- American Institute of Aeronautics and Astronautics (AIAA)
- Pennsylvania State University Alumni Associations

Select Public Publications

Weinert, A.; Erickson, P.; Reis, H.; Breimyer, P.; Hackett, T.; Samperi, M.; Huff, J.; Parra, C.; Stoekl, E.; Zundritsch, P.; Morris, R.; Iakimenko, I.; Petschauer, E.; Bilen, S., "Enabling communications in disadvantage environments: An airborne remote communication (ARC) platform," 2013 IEEE International Conference on Technologies for Homeland Security (HST), vol., no., pp.797,803, 12-14 Nov. 2013 doi: 10.1109/THS.2013.6699069

Erickson, P.; Weinert, A.; Breimyer, P.; Samperi, M.; Huff, J.; Parra, C.; Miller, S., "Designing public safety mobile applications for disconnected, interrupted, and low bandwidth communication environments," 2013 IEEE International Conference on Technologies for Homeland Security (HST), vol., no., pp.790,796, 12-14 Nov. 2013 doi: 10.1109/THS.2013.6699028

Weinert, A.J.; Breimyer, P.; Devore, S.M.; Miller, J.M.; Brulo, G.S.; Teal, R.S.; Zhang, D.; Kummer, A.T.; Bilen, S.G., "Providing communication capabilities during disaster response: Airborne remote communication (ARC) platform," 2012 IEEE Conference on Technologies for Homeland Security (HST), vol., no., pp.395,400, 13-15 Nov. 2012 doi: 10.1109/THS.2012.6459881

Griffith, J. D.; Edwards, M.; Miraflor, R.; and Weinert, A, Due Regard Encounter Model, Version 1.0. No. MIT/LL-ATC-397, Massachusetts Institute of Technology Lincoln Laboratory, 2013.

Maki, E.; Weinert, A.; and Kochenderfer, M., "Efficiently estimating ambient near mid-air collision risk for unmanned aircraft," AIAA Aviation Technology, Integration, and Operations Conference, Fort Worth, TX. 2010

In the News

- MIT Lincoln Lab researchers discuss games as tools for teaching, evaluating real-world skills, Polygon, 2013.
- Students hope to create more reliable emergency communications network, Penn State News, 2013
- Army Conducts Successful Demonstrations of the Ground Based Sense and Avoid System, Army.mil, 2012.